2023-2024-2 学期《数据结构》各考核环节成绩分段人数统计

考核环节		90-100分（A）	80-89分（B）	70-79分（C）	60-69分（D）	0-59分（E）	附原始资料样本份数
作业	作业一	60	23	35		2	A2 B2 C2
	作业二	62	14	43		1	A2 B2 C2
	作业三	55	37	26		2	A2 B2 C2
	作业四	54	28	36		2	A2 B2 C2
	作业五	56	35	28		1	A2 B2 C2
	作业六	56	28	35		1	A2 B2 C2
	作业七	54	30	35		1	A2 B2 C2
	作业八	51	29	19	20	1	A2 B2 C2 D2
实验	实验 1	79	23	17		1	A2 B2 C2
	实验 2	71	22	26		1	A2 B2 C2
	实验 3	75	23	20		2	A2 B2 C2
	实验 4	71	16	30		3	A2 B2 C2
	实验 5	67	23	19		11	A2 B2 C2 E2
	实验 6	70	21	25		4	A2 B2 C2
	实验 7	78	16	22		4	A2 B2 C2
	实验 8	68	15	33		4	A2 B2 C2
期中考试		14	69	23	8	6	A2 B2 C2 D2 E2
期末考试		15	19	14	23	49	A2 B2 C2 D2 E2

注：0-59 分（E）区间段未提供样本的原因是学生未提交作业或实验，成绩为 0 分。

目 录

1 作业样本……1

1.1 作业一……1

1.2 作业二…… 7

1.3 作业三……13

1.4 作业四……19

1.5 作业五……25

1.6 作业六……31

1.7 作业七……37

1.8 作业八……43

2 实验报告样本……51

2.1 实验报告 1……51

2.2 实验报告 2……108

2.3 实验报告 3……160

2.4 实验报告 4……220

2.5 实验报告 5……267

2.6 实验报告 6……339

2.7 实验报告 7……375

2.8 实验报告 8……406

3 期中考试答卷样本……435

4 期末考试答卷样本…… 451

(1) 算法原地工作的含义是指不需要任何额外的辅助空间

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1)，(2)　　C. (1)，(4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度(每小题 5 分，共 20 分)

```
1. for ( m = 0, i = 1; i <= n; i++)
     for (j = 1; j <= 2 * i; j++)
       m = m + 1;
```

答：$O(n^2)$

```
2. for (i = n- 1; i >= 1; i-- )
     for (j = 1; j <= i; j++)
       if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```

答：$O(n^2)$

```
3. for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
       a[i][j] = 0;
```

答：$O(mn)$

```
4. for (i = 1; i <= n; i++)
     if (2 * i <= n)
       for (j = 2 * i; j <= n; j++)
         y = y + i * j;
```

答：$O(n^2)$

三、算法设计题(每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法)

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

1. 伪代码

① 输入分子a和分母b

② 计算最大公约数 gcd(a,b)

a. 如果 b=0，则 gcd=a

b 否则，计算 gcd(b, a%b)

③ 化简分数

a. 分子 a = a/gcd.

b. 分母 b = b/gcd

④ 输出最简分数 a/b

C语言

```
#include<stdio.h>
int gcd(int a,int b)
{
  while (b!=0)
  { int temp=b;
    b=a%b;
    a=temp;}
  return a;
}
int main()
{ int a,b;
  scanf("%d %d",&a,&b);
  if (a>b)
  {printf("不是真分数\n");
   return 1;}
  int greatest_common_divisor=gcd(a,b);
  a/=greatest_common_divisor;
  b/=greatest_common_divisor;
  printf("最简分数为：%d %d\n",a,b);
  return 0;
}
```

2. 伪代码

① 函数 删除元素(数组r, 数组长度n, 目标值x);

② 初始化指针 j=0

③ 返回新长度 j

④ 遍历i 从0 到 n-1

⑤ 如果 r[i]!=x

r[j]=r[i]

j=j+1

(1) 算法原地工作的含义是指不需要任何额外的辅助空间是常数.

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1)，(2)　　C. (1)，(4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度（每小题 5 分，共 20 分）

$O(n)$

```
1. for ( m = 0, i = 1; i <= n; i++)
     for (j = 1; j <= 2 * i; j++)
       m = m + 1;
```

答：$O(n^2)$

```
2. for (i = n- 1; i >= 1; i-- )
     for (j = 1; j <= i; j++)
       if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```

$O(n)$　$\frac{n \times (n-1)}{2}$　$O(n^2)$

答：$O(n^2)$

```
3. for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
       a[i][j] = 0;
```

答：$O(mn)$

```
4. for (i = 1; i <= n; i++)
     if (2 * i <= n)
       for (j = 2 * i; j <= n; j++)
         y = y + i * j;
```

$O(n)$　$O(n)$

答：$O(n^2)$

三、算法设计题（每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法）

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

```
int DeleteAllx (int r[], int n, int x)
  { int k=0, i=0 ;
    for ( ; i<n; i++)
      {if (r[i]==x)
         k++;
      else r[i-k]=r[i] ; }
    return n-k;
```

1. 伪代码：

```
int gcd (a,b)
{ while (b!=0)
  { temp = b
    b= a%b
    a= temp }
  return a }
scanf("%d %d", &
m= m/gcd
n= n/gcd.
```

伪代码描述如下：

1. tempm=m ; tempn=n;
2. r=m%n
3. 重复下述操作直到r等于0:
   3.1 m=n; n=r
   3.2 r=m%n
4. 输出 tempm/n 和 tempn/n.

```
m=m/maxfactor ;
n= n/maxfactor;
int c = m/n;
return 0;
}.
```

伪代码：

```
remove (r, n, x)
=0 ;
i from 0 to n-1
if r[i]!=x
  r[a]= r[i]
  a++ }
return a
```

伪代码描述如下：

1. 初始化：k=0
2. 循环变量i从0~n-1
   2.1 如果 r[i] 等于x，则
   2.2 否则将 r[i] 前移 k
   2.3 i++;
3. 返回删除所有值为

(1) 算法原地工作的含义是指~~不需要任何额外~~的辅助空间是常数

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1)，(2)　　C. (1)，(4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度(每小题 5 分，共 20 分)

```
1. for ( m = 0, i = 1; i <= n; i++)
     for (j = 1; j <= 2 * i; j++)
       m = m + 1;
```

答：$O(n^2)$

```
2. for (i = n- 1; i >= 1; i-- )
     for (j = 1; j <= i; j++)
       if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```

答：$O(n^2)$

```
3. for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
       a[i][j] = 0;
```

答：$O(n)$

```
4. for (i = 1; i <= n; i++)
     if (2 * i <= n)
       for (j = 2 * i; j <= n; j++)
         y = y + i * j;
```

答：$O(n^2)$

三、算法设计题(每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法)

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

1.
```
#include
int gcd (int a, int b)
{while ( b!=0)
  { int temp=b;
      b = a%b
      a = temp; }
  return a; }
int main ( )
{scanf ( "%d %d",&m,&n);
 int maxfactor = gcd (m,n);
  m = m/maxfactor;
  n = n/maxfactor;
  int c = m/n ;
  return 0;
 }
```

2. C语言
```
void ChangeList(NODE *first)
{NODE *p=first , *q=first , *r=NULL , *s=NULL;
 while (q->next != NULL)
 { p=p->next ; q=q->next;
   if (q->next != NULL) q=q->next;
 } q = p->next;
 while (q != NULL)
   { r=q->next;
     q->next=p->next; p->next=q;
     q = r;
   }
 s= first->next;
 q = p->next;
 p->next =NULL;
 while (q != NULL)
 { r=q->next;
   q->next = s->next ; s->next=q;
   s=q->next ; q=r;
 }
 }
```

B

b. 否则，计算gcd (b, a%b)

③(x)简分数

a. 分子a=a/gcd

b. 分母b=b/gcd

①平衡八分子a和分母b

②计算最大公约数gcd(a,b)

a. 如果b=0，则gcd=a

(1) 算法原地工作的含义是指不需要任何额外的辅助空间

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1), (2)　　C. (1), (4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度（每小题 5 分，共 20 分）

```
1. for ( m = 0, i = 1; i <= n; i++)
     for (j = 1; j <= 2 * i; j++)
       m = m + 1;
```

答：$O(n)$　$O(n^2)$

```
2. for (i = n- 1; i >= 1; i-- )
     for (j = 1; j <= i; j++)
       if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```

答：$O(n^2)$

```
3. for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
       a[i][j] = 0;
```

答：$O(nm)$

```
4. for (i = 1; i <= n; i++)
     if (2 * i <= n)
       for (j = 2 * i; j <= n; j++)
         y = y + i * j;
```

答：$O(n^2)$

三、算法设计题（每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法）

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

答：1. ① 伪代码

输入分子a，分母b

d=gcd(a,b)　新分子=a/d

新分母=b/d

输出新分子/新分母

② C语言.

```
#include <stdio.h>
int gcd (int a, int b)
{
  while (b!=0)
  { int temp=b;
    b=a%b;
    a=temp; }
  return a; }
int main ( )
{
  int a,b;
  printf("请输入分子:");
  scanf("%d",&a);
  printf("请输入分母:");
  scanf("%d",&b);
  int d=gcd(a,b);
  a=a/d;
  b=b/d;
  printf("最简分数为:%d/%d\n", a,b);
  return 0;
}
```

2. ① 伪代码

定义变量 count=0

for i from 0 to n-1

if r[i]!=x

r[count]=count+1

新数组大小为 count

```
{ if (r[i]!=x)
  {r[count]=r[i];
   count++ }
}
}
```

② C语言.

```
#include <stdio.h>
void removeElement(int r[], int n, int x)
{ for(int i=0; i<n; i++)
int main( )
  { int r[]={1,2,3,2,4};
    int n=2; 2
    int n = sizeof(r)/sizeof(r[0]);
removeElement(r,n,x)
return 0;
}
```

(1) 算法原地工作的含义是指不需要任何额外的辅助空间

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1), (2)　　C. (1), (4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度(每小题 5 分，共 20 分)

1.
```
for ( m = 0, i = 1; i <= n; i++)
    for (j = 1; j <= 2 * i; j++)
      m = m + 1;
```
答：$O(n^2)$

2.
```
for (i = n- 1; i >= 1; i-- )
    for (j = 1; j <= i; j++)
      if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```
答：$O(n^2)$

3.
```
for (i = 0; i < n; i++)
    for (j = 0; j < m; j++)
      a[i][j] = 0;
```
答：$O(mn)$

4.
```
for (i = 1; i <= n; i++)
    if (2 * i <= n)
      for (j = 2 * i; j <= n; j++)
        y = y + i * j;
```
答：$O(n^2)$

三、算法设计题(每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法)

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

1. 伪代码.

① 输入分子 a, b.

② 计算最大公约数 gcd(a,b)

a. 如果 b=0，则 gcd=a

b. 否则，计算 gcd(b, a%b)

③ 化简分数

a. 分子 a=a/gcd

b. 分母 b=b/gcd.

④ 输出最简分数 a/b

(1) 算法原地工作的含义是指不需要任何额外的辅助空间

(2) 在相同的规模 n 下，复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法

(3) 所谓时间复杂度是指最坏情况下，估算算法执行时间的一个上界

(4) 同一个算法，实现语言的级别越高，执行效率就越低

A. (1)　　B. (1)，(2)　　C. (1)，(4)　　D. (3)

二、指出如下程序段的基本语句，并分析时间复杂度(每小题 5 分，共 20 分)

```
1. for ( m = 0, i = 1; i <= n; i++)
     for (j = 1; j <= 2 * i; j++)
       m = m + 1;
```

答：m=m+1

$O(n^2)$

```
2. for (i = n- 1; i >= 1; i-- )
     for (j = 1; j <= i; j++)
       if (A[j] > A[j+1]) 交换 A[j]与 A[j+1];
```

答：交换A[j]与A[j+1]

$O(n^2)$

```
3. for (i = 0; i < n; i++)
     for (j = 0; j < m; j++)
       a[i][j] = 0;
```

答：a[i][j]=0

$O(n^2)$

```
4. for (i = 1; i <= n; i++)
     if (2 * i <= n)
       for (j = 2 * i; j <= n; j++)
         y = y + i * j;
```

答：y=y+i*j

$O(n^2)$

三、算法设计题(每小题 20 分，共 40 分，用伪代码和 C 语言两种方式描述算法)

1. 分式化简。将一个给定的真分数化简为最简分数形式。例如，将 6/8 化简为 3/4。

2. 在整型数组 r[n]中删除所有值为 x 的元素，要求时间复杂度为 $O(n)$，空间复杂度为 $O(1)$。

1, 伪：输入两个自然数 m和n

输出以m为分子 n为分母的分数最简式

1, r=m%n;

2 a=m;

3, b=n

4, r到

4.1 m=n

4.2 n=r;

4.3 r=m%n;

5. x=a/n;

6. y=b/n

7 输出x/y

2, 伪代码

输入 n和k

输出数;

1, i=0;

2, i=n

2.1 计(A[i]=k)

break;

2.2 计+;

3, 输出;;

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是(　　)。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;

A. (1)(2)(3)(4)　　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geq n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

```
1) bool isIncreasing(ListNode * head)
if (head==NULL || head->next==NULL)
return true;
ListNode *p = head;
while (p->next!=NULL)
{ if (p->data>p->next->data)
    return false;
  p = p->next; }
return true;
```

```
2) void deletePredecessor(ListNode *s)
{ ListNode * pre = s;
while (pre->next->next != s)
    { pre = pre->next; }
ListNode * toDelete = pre->next; pre->next=s;
free(toDelete); }
```

```
3) void duplicateList(ListNode * head)
{ ListNode *p = head;
  while (p!=NULL)
{ ListNode * newNode = (ListNode *)
  malloc(sizeof(ListNode));
  newNode->data = p->data;
  newNode->next = p->next;
    p->next = newNode;
    p= p->next->next; }
}
```

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

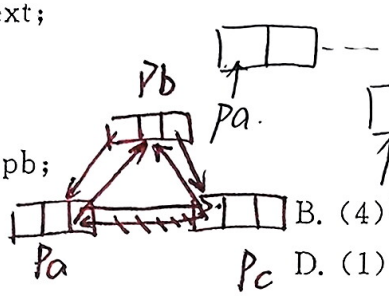
10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是（ D ）。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;



A. (1)(2)(3)(4)　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geq n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

```
1. bool isIncreasing(ListNode *head)
   if(head==NULL||head->next==NULL)
     return true;
   ListNode *p=head;
   while(p->next!=NULL)
   { if(p->data>p->next->data)
       return false;
     p=p->next;}
   return true;

2. Void deletePredecessor(ListNode *s)
   { ListNode *pre=s;
     While(pre->next->next!=s)
       { pre=pre->next;}
     ListNode *toDelete=pre->next;
     pre->next=s;
     free(toDelete);}

3. Void duplicatelist(ListNode *head)
   { ListNode *p=head;
     while(p!=NULL)
     { ListNode *newNode=(ListNode*)
       malloc(sizeof(ListNode));
       newNode->data=p->data;
       newNode->next=p->next;
       p->next=newNode;
       p=p->next->next;}}
```

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是（ ）。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;

A. (1)(2)(3)(4)　　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geqslant n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

解：1.
```
bool increase(LinkList *first)
{Node *p, *q;
 p=first->next;
 while(p->next)
 { q=p->next;
   if(p->data<q->data)
   p=q;
   else return 0;}
   return 1;}
```

2.
```
template<class T>
Void Del(Node<T> *s)
{ p=s;
  while(p->next->next!=s)
   p=p->next;
   p=p->next;
  p->next=s;
  delete r;
}
```

3.
```
Void duplicateList(ListNode *head)
{ ListNode *p=head;
  while(p!=NULL)
{ ListNode * newNode=(List Node *)
  malloc(sizeof(ListNode));
  newNode->data=p->data;
  newNode->next=p->next;
  p->next=newNode;
  p=p->next->next;}
}
```

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是(　　)。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;

A. (1)(2)(3)(4)　　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geqslant n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

```
1. bool increase (LinkList *first)
{ Node *p, *q;
  p=first->next;
  while (p->next)
  { q=p->next;
    if (p->data>q->data) p=q;
    else return 0;
  }
  return 1;
}
```

```
2. void del(LNode<T> *s)
{ p=s;
  while (p->next->next!=s)
    p=p->next;
  r=p->next;
  p->next=s;
  delete r;
}
```

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是(　　)。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;

A. (1)(2)(3)(4)　　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geq n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

```
1. bool increase (LinkList *first)
{ Node *p,*q;
  p=first->next;
  while(p->next)
    { q=p->next;
      if (p->data<q->data) p=q;
      else return 0;
    }
  return 1;
}
```

```
2. template<class T>
void Del(Node<T>*s)
{
  p=s;
  while (p->next->next!=s)
      p=p->next;
  r=p->next;
  p->next=s;
  delete r;
}
```

```
3. void doubleList
(Node *&head)
{ Node *p=head;
  while (p!=nullptr)
    Node * newNode =new
Node(p->data);
    newNode->next=p->
    p->next=newNode;
    next->next;
```

A. rear=rear->next;

B. s=rear->next; rear->next=rear->next->next;

C. rear=rear->next->next;

D. s=rear->next->next; rear->next->next=s->next;

10. 在双链表中指针 pa 所指结点后面插入 pb 所指结点，执行的语句序列是(　　)。

(1) pb->next=pa->next;

(2) pb->prior=pa;

(3) pa->next=pb;

(4) pa->next->prior=pb;

A. (1)(2)(3)(4)　　B. (4)(3)(2)(1)

C. (3)(4)(1)(2)　　D. (1)(4)(3)(2)

二、回答下列问题(每小题 10 分，共 30 分)

1. 设 n 表示线性表的元素个数，E 表示存储数据元素所需的存储单元大小，D 表示数组中存储线性表的最大元素个数($D \geqslant n$)，采用顺序存储方式存储该线性表需要多少存储空间？设 P 表示指针所需的存储单元大小，使用单链表存储方式存储该线性表需要多少存储空间(不考虑头结点)？

2. 如果某线性表中数据元素的类型不一致，但是希望能根据下标随机存取每个元素，请为这个线性表设计一个合适的存储结构。

3. 带头结点的单链表和不带头结点的单链表(假设头指针是 first)为空的判断条件是什么？分别画出存储示意图。

三、算法设计(共 40 分)

1. 判断单链表是否递增有序(12 分)。

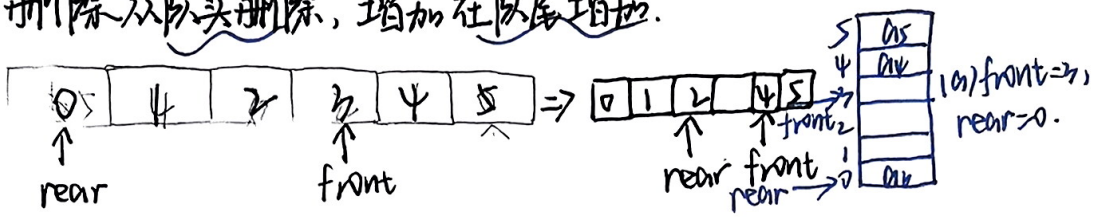
2. 假设在长度大于 1 的循环单链表中，即无头结点也无头指针，s 为指向链表中某个结点的指针，要求删除结点 s 的前趋结点(12 分)。

3. 以单链表作为存储结构，将线性表扩展 2 倍，假设线性表为(a_1, a_2, …, a_n)，扩展后为(a_1, a_1, a_2, a_2, …, a_n, a_n)(16 分)。

```
1. int isIncreasing(struct ListNode* head)
{ struct ListNode *p=head;
  while(p!=NULL && p->next!=NULL)
    { if(p->data > p->next->data)
      { return 0;
      }
    p=p->next;
    }
  return 1;
}
```

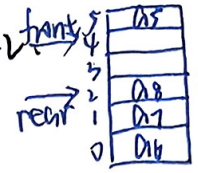
```
2. void deletePredecessor(struct ListNode* s)
{ if(s==NULL||s->next==s)
    { return;}
  struct ListNode * prepre=s;
  struct ListNode* pre=s->next;
  while(pre->next!=s){
      prepre=pre;
      pre=pre->next;
  }
  prepre->next=s;
  free(pre);
}
```

1. 删除从队头删除，增加在队尾增加.



∵ front = (7+1)%8 = 4
rear = (0+1)%8 = 1
rear = (1+1)%8 = 2

∵ front=4, rear=2.

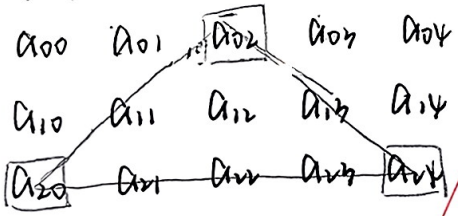


2. 用count记录栈中元素个数，初始值为0；当遇到1时，count++；
当遇到0时，count--。当最终count=0，则操作合法。count始终>0.

例：1100011：
①遇1，count=1
②遇1；count=2
③遇0：count=1
④遇0：count=0
⑤遇0：count=-1
∴该序列非法.

1100101
①遇1：count=1
②遇1：count=2
③遇0：count=1
④遇0：count=0
⑤遇1：count=1
⑥遇0：count=0
∴该序列合法.

4. 设n=3，2n-1=5.



如图所示，第0行一个元素，
第1行三个元素，第2行五个.
∴第n-1行：2n-1个.
∴a[i][j]前有i行
1+3+5+...+2(i-1)-1
=1+3+5+...+(2i-1)
$=\frac{(2i-1)\cdot i}{2}=\frac{i^2+i}{2}=i^2+i$

∴k = i²+i+1+j+1
= i²+i+j+2.

7. 根据运算符优先级确定运算次序为
乘法和除法的优先级高于加、减法.
因此先计算B*C，接着计算B*C/D
接着计算减法：A-(B*C/D)
接着进行除法运算，最后，计算加法部分

(A-(B*C/D))+(E/F)

(A-(B*C/D))+E/F

栈的作用：
①操作数栈：将A、B、C、D等依次存入栈中
②操作符栈：

C. $\frac{j(j+1)}{2}+i$　　D. $\frac{j(j-1)}{2}+i-1$

10. 有一个 100 * 90 的整型稀疏矩阵，其中非零元素个数为 10，设整型数据占 2 字节，则采用三元组顺序表存储时，需要的字节数是（ B ）。

A. 60　　B. 66　　C. 18 000　　D. 33

11. 假设用数组 A[21]存储循环队列，front 指向队头元素的前一个位置，rear 指向队尾元素，假设当前 front 和 rear 的值分别为 8 和 3，则该队列的长度为（ C ）。

A. 5　　B. 6　　C. 16　　D. 17

12. 在栈顶指针为 top 的带头结点的链栈中，插入指针 s 所指的结点，应执行（ D ）。

A. top->next=s;　　B. s->next=top;

C. s->next=top; top->next=s　　D. s->next=top->next; top->next=s;

13. 从栈顶指针为 top 的不带头结点的链栈中删除一个结点，用 x 保存被删除结点的值，则执行（ D ）。

A. x=top; top=top->next;

B. x=top->data;

C. top=top->next; x=top->data;

D. x=top->data; top=top->next;

14. 循环队列存储在数组 A[0..m]中，则入队时的操作为（ D ）。

A. rear=rear+1　　B. rear=(rear+1) mod (m−1)

C. rear=(rear+1) mod m　　D. rear=(rear+1) mod (m+1)

15. 在链队列中，设指针 f 和 r 分别指向队首和队尾，则插入 s 所指结点的操作是（ B ）。

A. f->next=s; f=s;　　B. r->next=s; r=s;

C. s->next=r; r=s;　　D. s->next=f; f=s;

二、回答下列问题(每小题 10 分，共 40 分)

1. 若用一个长度为 6 的数组实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，则从队列中删除一个元素，再增加两个元素后，rear 和 front 的值分别是多少？请画出操作示意图。

front = (3+1)%6=4　　rear = (0+2)%6=2.

2. 假设 I 和 O 分别表示入栈和出栈操作。栈的初态和终态均为空，入栈和出栈的操作序列可表示为仅由 I 和 O 组成的序列，称可以操作的序列为合法序列，否则称为非法序列。如何判定所给的操作序列是否合法呢？请结合具体实例给出判定规则。

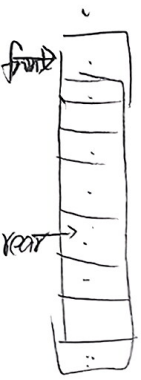
3. 给出表达式 $A-B*C/D+E/F$ 的求值过程，说明在求值过程中栈的作用。

4. 对于给定的数组 a[n][2n−1]，非零元素在顶点为 a[0][n−1]、a[n−1][0]和 a[n−1][2n−2]的三角形内(包括三角形的边)，三角形以外的元素值均为零。将所有非零元素存储在一维数组 B[n×n]中，请给出压缩存储方案。如果位于三角形内的元素 a[i][j]存放于 B[k]中，请给出元素 a[i][j]的下标 i，j 与 k 之间的对应关系。

C. $\frac{j(j+1)}{2}+i$　　D. $\frac{j(j-1)}{2}+i-1$

10. 有一个 100 * 90 的整型稀疏矩阵，其中非零元素个数为 10，设整型数据占 2 字节，则采用三元组顺序表存储时，需要的字节数是（　　）。

A. 60　　B. 66　　C. 18 000　　D. 33



11. 假设用数组 A[21]存储循环队列，front 指向队头元素的前一个位置，rear 指向队尾元素，假设当前 front 和 rear 的值分别为 8 和 3，则该队列的长度为（　　）。

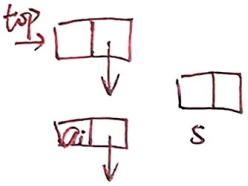
A. 5　　B. 6　　C. 16　　D. 17

12. 在栈顶指针为 top 的带头结点的链栈中，插入指针 s 所指的结点，应执行（　　）。

A. top->next=s;　　B. s->next=top;

C. s->next=top; top->next=s　　D. s->next=top->next; top->next=s;

13. 从栈顶指针为 top 的不带头结点的链栈中删除一个结点，用 x 保存被删除结点的值，则执行（　　）。



A. x=top; top=top->next;

B. x=top->data;

C. top=top->next; x=top->data;

D. x=top->data; top=top->next;

14. 循环队列存储在数组 A[0..m]中，则入队时的操作为（　　）。

A. rear=rear+1　　B. rear=(rear+1) mod (m−1)

C. rear=(rear+1) mod m　　D. rear=(rear+1) mod (m+1)

15. 在链队列中，设指针 f 和 r 分别指向队首和队尾，则插入 s 所指结点的操作是（　　）。

B. r->next=s; r=s;

D. s->next=f; f=s;

分，共 40 分）

现循环队列，且当前 rear 和 front 的值分别为 0 和 3，

两个元素后，rear 和 front 的值分别是多少？请画出操

出栈操作。栈的初态和终态均为空，入栈和出栈的操

序列，称可以操作的序列为合法序列，否则称为非法

否合法呢？请结合具体实例给出判定规则。

E/F 的求值过程，说明在求值过程中栈的作用。

]，非零元素在顶点为 a[0][n−1]、a[n−1][0]和 a[n−1]

的边），三角形以外的元素值均为零。将所有非零元素存储

压缩存储方案。如果位于三角形内的元素 a[i][j]存放于 B

[k]中，请给出元素a[i][j]的下标 i，j 与 k 之间的对应关系。

C. $\frac{j(j+1)}{2}+i$　　D. $\frac{j(j-1)}{2}+i-1$

10. 有一个 100 * 90 的整型稀疏矩阵，其中非零元素个数为 10，设整型数据占 2 字节，则采用三元组顺序表存储时，需要的字节数是（　）。

A. 60　　B. 66　　C. 18 000　　D. 33

11. 假设用数组 A[21]存储循环队列，front 指向队头元素的前一个位置，rear 指向队尾元素，假设当前 front 和 rear 的值分别为 8 和 3，则该队列的长度为（　）。

A. 5　　B. 6　　C. 16　　D. 17

12. 在栈顶指针为 top 的带头结点的链栈中，插入指针 s 所指的结点，应执行（　）。

A. top->next=s;　　B. s->next=top;

C. s->next=top; top->next=s　　D. s->next=top->next; top->next=s;

13. 从栈顶指针为 top 的不带头结点的链栈中删除一个结点，用 x 保存被删除结点的值，则执行（　）。

A. x=top; top=top->next;

B. x=top->data;

C. top=top->next; x=top->data;

D. x=top->data; top=top->next;

14. 循环队列存储在数组 A[0..m]中，则入队时的操作为（　）。

A. rear=rear+1　　B. rear=(rear+1) mod (m−1)

C. rear=(rear+1) mod m　　D. rear=(rear+1) mod (m+1)

15. 在链队列中，设指针 f 和 r 分别指向队首和队尾，则插入 s 所指结点的操作是（　）。

A. f->next=s; f=s;　　B. r->next=s; r=s;

C. s->next=r; r=s;　　D. s->next=f; f=s;

二、回答下列问题(每小题 10 分，共 40 分)

1. 若用一个长度为 6 的数组实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，则从队列中删除一个元素，再增加两个元素后，rear 和 front 的值分别是多少？请画出操作示意图。

2. 假设 I 和 O 分别表示入栈和出栈操作。栈的初态和终态均为空，入栈和出栈的操作序列可表示为仅由 I 和 O 组成的序列，称可以操作的序列为合法序列，否则称为非法序列。如何判定所给的操作序列是否合法呢？请结合具体实例给出判定规则。

3. 给出表达式 $A-B*C/D+E/F$ 的求值过程，说明在求值过程中栈的作用。

4. 对于给定的数组 a[n][2n−1]，非零元素在顶点为 a[0][n−1]、a[n−1][0]和 a[n−1][2n−2]的三角形内(包括三角形的边)，三角形以外的元素值均为零。将所有非零元素存储在一维数组 B[n×n]中，请给出压缩存储方案。如果位于三角形内的元素 a[i][j]存放于 B[k]中，请给出元素 a[i][j]的下标 i，j 与 k 之间的对应关系。

C. $\frac{j(j+1)}{2}+i$　　D. $\frac{j(j-1)}{2}+i-1$

10. 有一个 100 * 90 的整型稀疏矩阵，其中非零元素个数为 10，设整型数据占 2 字节，则采用三元组顺序表存储时，需要的字节数是（　　）。

A. 60　　B. 66　　C. 18 000　　D. 33

11. 假设用数组 A[21]存储循环队列，front 指向队头元素的前一个位置，rear 指向队尾元素，假设当前 front 和 rear 的值分别为 8 和 3，则该队列的长度为（　　）。

A. 5　　B. 6　　C. 16　　D. 17

12. 在栈顶指针为 top 的带头结点的链栈中，插入指针 s 所指的结点，应执行（　　）。

A. top->next=s;　　B. s->next=top;

C. s->next=top; top->next=s　　D. s->next=top->next; top->next=s;

13. 从栈顶指针为 top 的不带头结点的链栈中删除一个结点，用 x 保存被删除结点的值，则执行（　　）。

A. x=top; top=top->next;

B. x=top->data;

C. top=top->next; x=top->data;

D. x=top->data; top=top->next;

14. 循环队列存储在数组 A[0..m]中，则入队时的操作为（　　）。

A. rear=rear+1　　B. rear=(rear+1) mod (m-1)

C. rear=(rear+1) mod m　　D. rear=(rear+1) mod (m+1)

15. 在链队列中，设指针 f 和 r 分别指向队首和队尾，则插入 s 所指结点的操作是（　　）

A. f->next=s; f=s;　　B. r->next=s; r=s;

C. s->next=r; r=s;　　D. s->next=f; f=s;

二、回答下列问题(每小题 10 分，共 40 分)

1. 若用一个长度为 6 的数组实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，则从队列中删除一个元素，再增加两个元素后，rear 和 front 的值分别是多少？请画出操作示意图。

2. 假设 I 和 O 分别表示入栈和出栈操作。栈的初态和终态均为空，入栈和出栈的操作序列可表示为仅由 I 和 O 组成的序列，称可以操作的序列为合法序列，否则称为非法序列。如何判定所给的操作序列是否合法呢？请结合具体实例给出判定规则。

3. 给出表达式 $A-B*C/D+E/F$ 的求值过程，说明在求值过程中栈的作用。

4. 对于给定的数组 a[n][2n-1]，非零元素在顶点为 a[0][n-1]、a[n-1][0]和 a[n-1][2n-2]的三角形内(包括三角形的边)，三角形以外的元素值均为零。将所有非零元素存储在一维数组 B[n×n]中，请给出压缩存储方案。如果位于三角形内的元素 a[i][j]存放于 B[k]中，请给出元素 a[i][j]的下标 i，j 与 k 之间的对应关系。

C. $\frac{j(j+1)}{2}+i$　　D. $\frac{j(j-1)}{2}+i-1$

10. 有一个 100 * 90 的整型稀疏矩阵，其中非零元素个数为 10，设整型数据占 2 字节，则采用三元组顺序表存储时，需要的字节数是（ ）。

A. 60　　B. 66　　C. 18 000　　D. 33

11. 假设用数组 A[21]存储循环队列，front 指向队头元素的前一个位置，rear 指向队尾元素，假设当前 front 和 rear 的值分别为 8 和 3，则该队列的长度为（ ）。

A. 5　　B. 6　　C. 16　　D. 17

12. 在栈顶指针为 top 的带头结点的链栈中，插入指针 s 所指的结点，应执行（ ）。

A. top->next=s;　　B. s->next=top;

C. s->next=top; top->next=s　　D. s->next=top->next; top->next=s;

13. 从栈顶指针为 top 的不带头结点的链栈中删除一个结点，用 x 保存被删除结点的值，则执行（ ）。

A. x=top; top=top->next;

B. x=top->data;

C. top=top->next; x=top->data;

D. x=top->data; top=top->next;

14. 循环队列存储在数组 A[0..m]中，则入队时的操作为（ ）。

A. rear=rear+1　　B. rear=(rear+1) mod (m-1)

C. rear=(rear+1) mod m　　D. rear=(rear+1) mod (m+1)

15. 在链队列中，设指针 f 和 r 分别指向队首和队尾，则插入 s 所指结点的操作是（ ）。

A. f->next=s; f=s;　　B. r->next=s; r=s;

C. s->next=r; r=s;　　D. s->next=f; f=s;

二、回答下列问题(每小题 10 分，共 40 分)

1. 若用一个长度为 6 的数组实现循环队列，且当前 rear 和 front 的值分别为 0 和 3，则从队列中删除一个元素，再增加两个元素后，rear 和 front 的值分别是多少？请画出操作示意图。

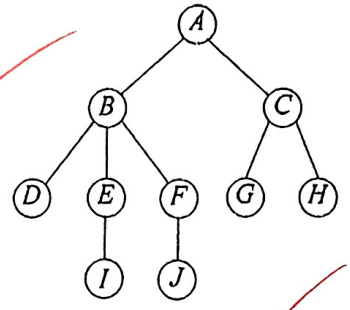
2. 假设 I 和 O 分别表示入栈和出栈操作。栈的初态和终态均为空，入栈和出栈的操作序列可表示为仅由 I 和 O 组成的序列，称可以操作的序列为合法序列，否则称为非法序列。如何判定所给的操作序列是否合法呢？请结合具体实例给出判定规则。

3. 给出表达式 $A-B*C/D+E/F$ 的求值过程，说明在求值过程中栈的作用。

4. 对于给定的数组 a[n][2n-1]，非零元素在顶点为 a[0][n-1]、a[n-1][0]和 a[n-1][2n-2]的三角形内(包括三角形的边)，三角形以外的元素值均为零。将所有非零元素存储在一维数组 B[n×n]中，请给出压缩存储方案。如果位于三角形内的元素 a[i][j]存放于 B[k]中，请给出元素 a[i][j]的下标 i，j 与 k 之间的对应关系。

二、回答下列问题(每个题 10 分,共 30 分)

1. 对于下图所示树结构:(1)画出双亲表示法存储示意图;(2)画出孩子表示法存储示意图;(3)画出孩子兄弟表示法存储示意图。



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *CDBAFEHG*,请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*,中序遍历序列为 *DBGEHACIF*,请构造该二叉树。

三、算法设计(共 40 分)

1. 假设二叉树以二叉链表存储,请在内存中复制该二叉链表(10 分)。

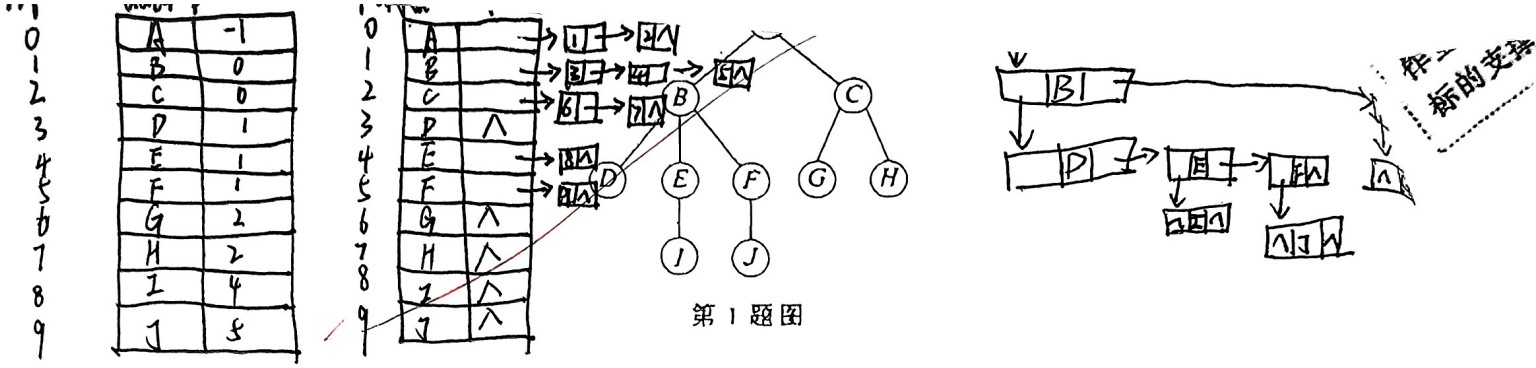
2. 设二叉树有 $n(n\leqslant 100)$ 个结点,按顺序存储方式存储在数组 bt[1..n]中,求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

3. 假设二叉树以二叉链表存储,求二叉树第 $k(k>1)$ 层上叶子结点的个数(18 分)。

```
1. #include <stdio.h>
   #include <stdlib.h>
   typedef struct BiTNode
   {int data;
   struct BiTNode *lchild,*rchild;
   }BiNode, *BiTree;
   BiTree CopyBiTree(BiTree T)
   {if(T==NULL);
   BiTree newNode=(BiTree)malloc(
   (sizeof(BiTree));
   if(newNode==NULL)
   {return NULL;}
   newNode->lchild=copyBiTree(T->lchild);
   newNode->rchild=copyBiTree(T->rchild);
   return newNode;}
```

```
2. #include <stdio.h>
   int lowest_common_A(int bt[], int i, int j)
   {while(i!=j)
   {if(i>j){i=i/2;}
     else{i=j/2;}}
   return bt[i];}
```

```
3. #include <stdio.h>
   #include <stdlib.h>
   typedef struct BiNode
   {int data;
   struct BiTNode *lchild,
   *rchild; BiTNode, *BiTree;
   int leafcount_(BiTree root, int k)
   {if(root==NULL||k<1)
     {return 0;}
   BiTree queue[100];
   int front=0, rear=0;
   queue[rear++]=root;
   int & level=1;
   int count=0;
   while(front<rear)
   {int size=rear-front;
   for(int i=0; i<size; i++)
   {BiTree node=queue[front++];
   if(level==k&&node->lchild==
   NULL&&node->rchild==NULL
   {count++;}
   if(level==k)
   {break;}
   level++;
   return count;}
```



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *CDBAFEHG*，请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*，中序遍历序列为 *DBGEHACIF*，请构造该二叉树。

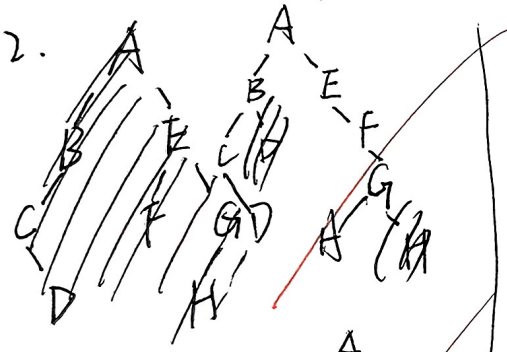
三、算法设计(共 40 分)

1. 假设二叉树以二叉链表存储，请在内存中复制该二叉链表(10 分)。

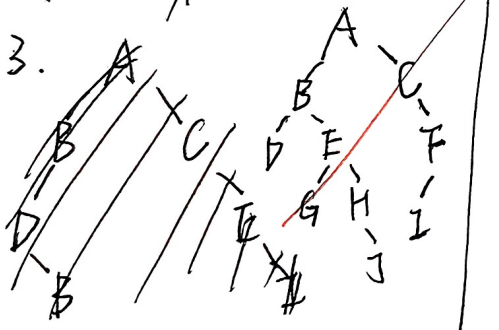
2. 设二叉树有 $n(n\leqslant 100)$ 个结点，按顺序存储方式存储在数组 bt[1..n]中，求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

3. 假设二叉树以二叉链表存储，求二叉树第 $k(k>1)$ 层上叶子结点的个数(18 分)。

二. 2.



3.



三、1.

```
typedef struct BiTNode{
    char data;
    struct BiTNode *lchild, *rchild;}
BiTNode, *BiTree;
BiTree CopyBiTree(BiTree original){
    if (original==NULL){
        return NULL;}
    BiTree newNode=(BiTree)malloc(sizeof(BiTNode));
newNode->data=original->data;
    newNode->lchild=
    CopyBiTree(original->lchild);
    newNode->rchild=
CopyBiTree(original->rchild);
        return newNode;}
```

2.

```
int FindLCA(int b[], int n, int i, int j){
  if(i<1||i>n||j<1||j>n){
        return -1;}
while(i!=j){
  if(i>j){
     i=i/2;}
  else{
     j=j/2;}
  }
return i;
}
```

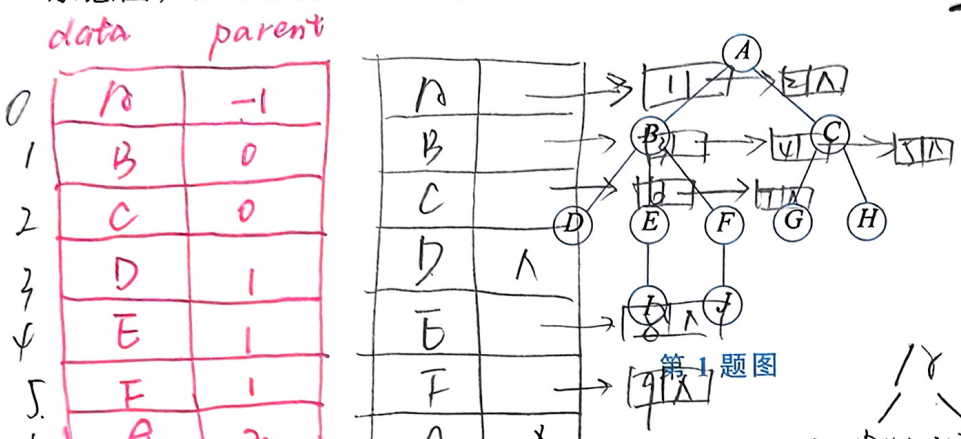
3.

```
int CountLevelLeaf(BiTree T, int k)
{ if(T==NULL||k<1){
        return 0;}
    if(k==1){
    return(T->lchild==NULL&&
  T->rchild==NULL){1:0;}
return CountLevelLeaf(T->lchild,k-1)+
CountLevelLeaf(T->rchild, k-1);}
```

扫描全能王
3亿人都在用的扫描App

二、回答下列问题（每个题 10 分，共 30 分）

1. 对于下图所示树结构：(1)画出双亲表示法存储示意图；(2)画出孩子表示法存储示意图；(3)画出孩子兄弟表示法存储示意图。



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *CDBAFEHG*，请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*，中序遍历序列为 *DBGEHACIF*，请构造该二叉树。

三、算法设计（共 40 分）

1. 假设二叉树以二叉链表存储，请在内存中复制该二叉链表(10 分)。

2. 设二叉树有 $n(n\leqslant 100)$ 个结点，按顺序存储方式存储在数组 bt[1..n]中，求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

3. 假设二叉树以二叉链表存储，求二叉树第 $k(k>1)$ 层上叶子结点的个数(18 分)。

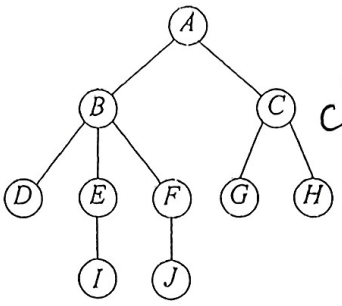
```
T1.
root->data = tree->data;
root->lchild = root->rchild
    =NULL;
if(p->lchild)
if(!visit(p->lchild->data))
return error;
newtree->data = p->lchild->data;
newtree->lchild = newtree->rchild
    =NULL;
if(p=tree) root->lchild = newtree;
else if(p->lchild==p->parent->lchild)
p->parent->lchild->rchild=newtree;
```

```
T2. int Ancestor(ElemType A[], int n, int i, int j)
while(i!=j) if(i>j)
    i=i/2;
else
    j=j/2;
return i;
```

```
T3. int front=0, rear=1, leaf=0;
int last=1, level=1; Q[1]=p;
if(p==null) return 0;
while(front<=rear)
{ p=Q[++front];
  if(level==k&&!p->lchild&&!p->rchild)leaf++;
  if(p->lchild) Q[++rear]=p->lchild;
  if(p->rchild) Q[++rear]=p->rchild;
  if(front==last){level++; last=rear;}
}
return 0;
```

二、回答下列问题(每个题 10 分,共 30 分)

1. 对于下图所示树结构:(1)画出双亲表示法存储示意图;(2)画出孩子表示法存储示意图;(3)画出孩子兄弟表示法存储示意图。



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *CDBAFEHG*,请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*,中序遍历序列为 *DBGEHACIF*,请构造该二叉树。

三、算法设计(共 40 分)

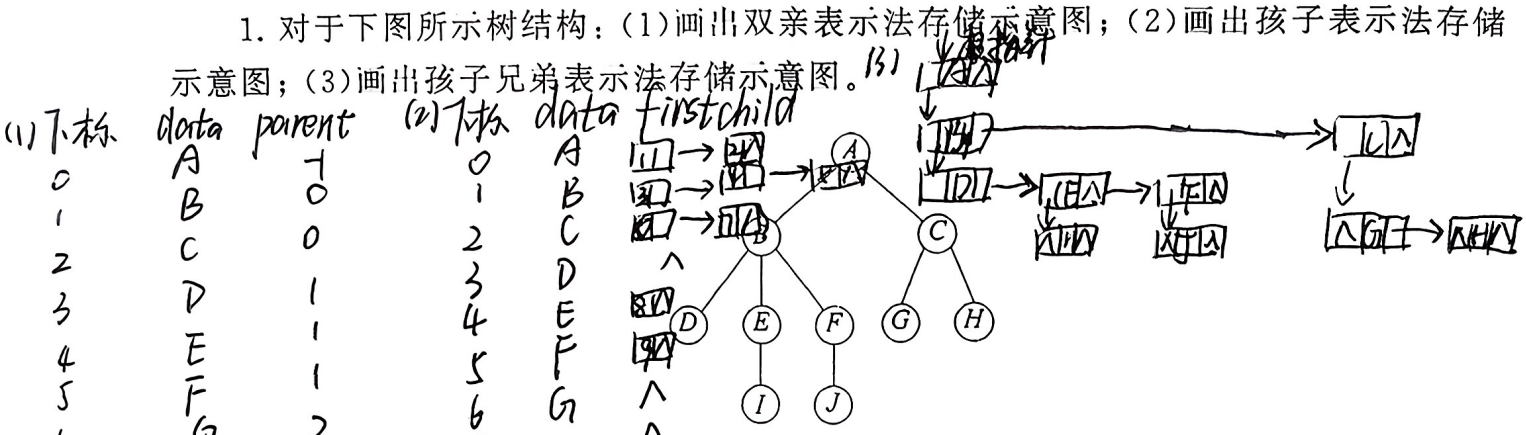
1. 假设二叉树以二叉链表存储,请在内存中复制该二叉链表(10 分)。

2. 设二叉树有 $n(n\leqslant 100)$ 个结点,按顺序存储方式存储在数组 bt[1..n]中,求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

3. 假设二叉树以二叉链表存储,求二叉树第 $k(k>1)$ 层上叶子结点的个数(18 分)。

二、回答下列问题(每个题 10 分,共 30 分)

1. 对于下图所示树结构:(1)画出双亲表示法存储示意图;(2)画出孩子表示法存储示意图;(3)画出孩子兄弟表示法存储示意图。



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *CDBAFEHG*,请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*,中序遍历序列为 *DBGEHACIF*,请构造该二叉树。

三、算法设计(共 40 分)

1. 假设二叉树以二叉链表存储,请在内存中复制该二叉链表(10 分)。

2. 设二叉树有 $n(n\leqslant 100)$ 个结点,按顺序存储方式存储在数组 bt[1..n]中,求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

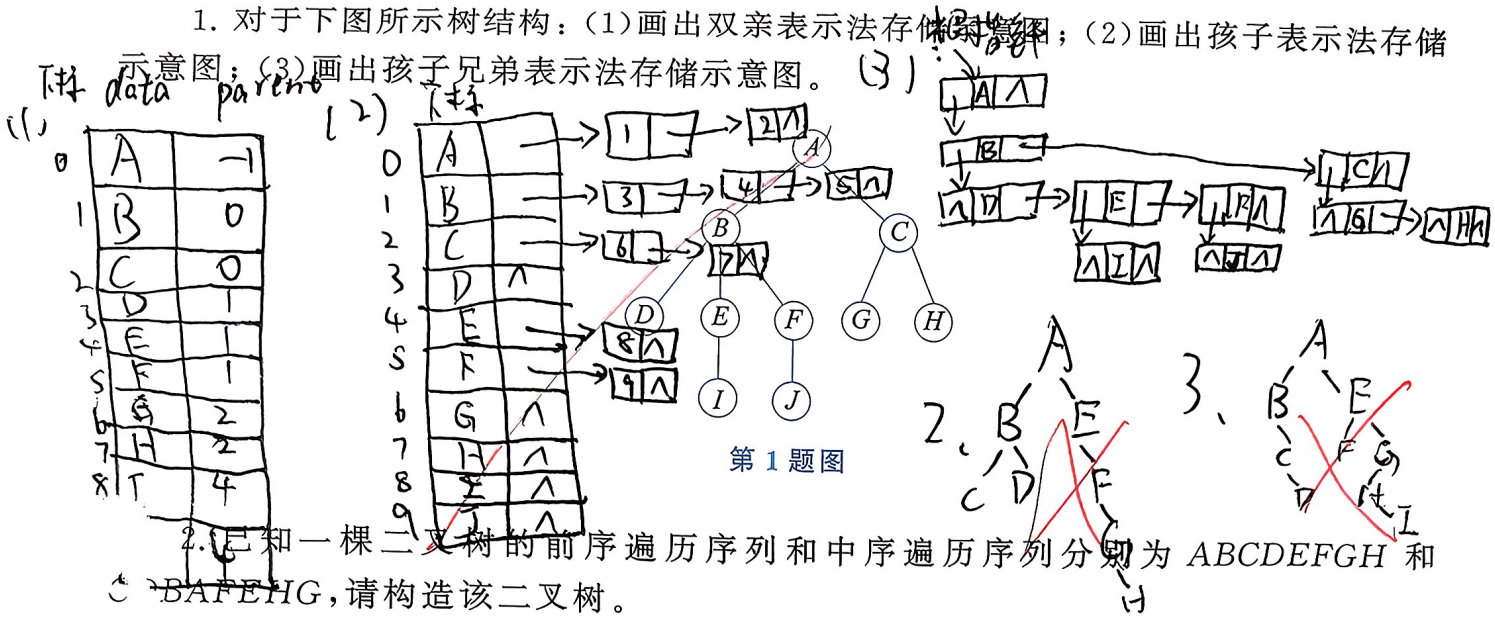
3. 假设二叉树以二叉链表存储,求二叉树第 $k(k>1)$ 层上叶子结点的个数(18 分)。

1.
```
#include<stdio.h>
#include<stdio.h>
typedef struct BiTNode
{ int data;
  struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
  BiTree copyBiTree (BiTree T)
{ if(T==NUIL)
{ return Null; }
BiTree newNode = (BiTree) malloc
(size of (BiTree));
if(newNode == NULL)
{return Null;}
new Node -> data = T -> data;
new Node -> lchild = copyBiTree(T->lchild);
new Node -> rchild = copyBiTree(T->rchild);
return new Node;}
```

2.
```
#include<stdio.h>
int lowest CommonA(int bt[],
int i, int j)
{while (i!=j)
{ if(i>j) i=i/2; )
  else j=j/2;}
return bt[i];}
```

二、回答下列问题(每个题 10 分,共 30 分)

1. 对于下图所示树结构:(1)画出双亲表示法存储示意图;(2)画出孩子表示法存储示意图;(3)画出孩子兄弟表示法存储示意图。



第 1 题图

2. 已知一棵二叉树的前序遍历序列和中序遍历序列分别为 *ABCDEFGH* 和 *BAFEHG*,请构造该二叉树。

3. 已知一棵二叉树的层序遍历序列为 *ABCDEFGHI*,中序遍历序列为 *DBGEHACIF*,请构造该二叉树。

三、算法设计(共 40 分)

1. 假设二叉树以二叉链表存储,请在内存中复制该二叉链表(10 分)。

2. 设二叉树有 $n(n\leqslant 100)$ 个结点,按顺序存储方式存储在数组 bt[1..n]中,求二叉树中编号为 i 和 j 的两个结点的最近公共祖先结点(12 分)。

3. 假设二叉树以二叉链表存储,求二叉树第 $k(k\geqslant 1)$ 层上叶子结点的个数(18 分)。

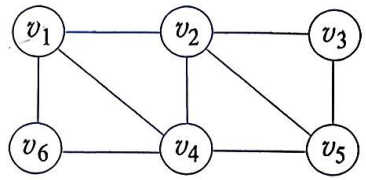
到的不同遍历序列个数是（D）。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题（每小题 10 分，共 40 分）

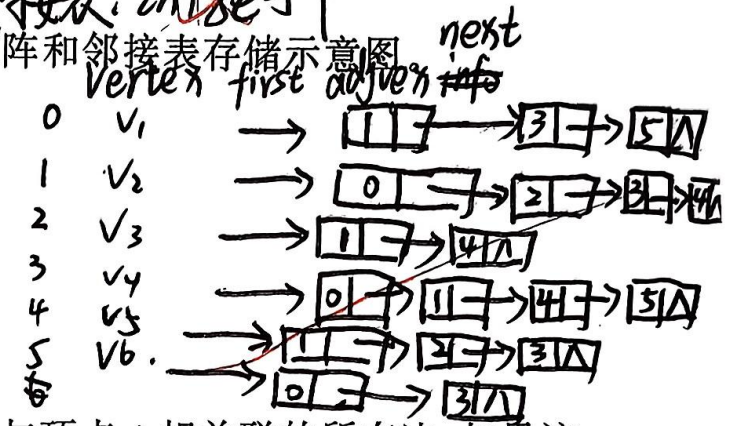
1. 设无向网图 G 含有 n 个顶点 e 条边，每个顶点的信息（假设只存储编号）占用 2 字节，每条边的权值信息占用 4 字节，每个指针占用 4 字节，计算采用邻接矩阵和邻接表分别占用多少存储空间？邻接矩阵：4n²+2n字节　邻接表：2n+8e字节

2. 已知一个连通图如下图所示，试给出图的邻接矩阵和邻接表存储示意图。



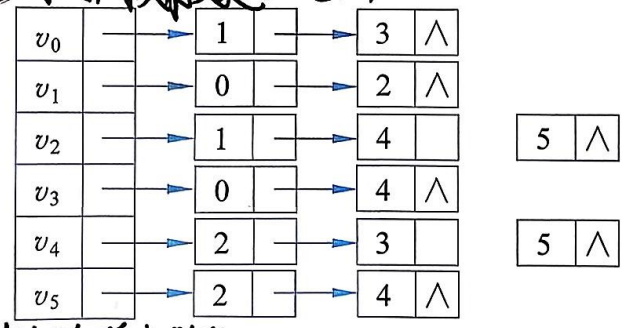
第 2 题图

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	0	1	0	1	0	1
V_2	1	0	1	1	1	0
V_3	0	1	0	0	1	0
V_4	1	1	0	0	1	1
V_5	0	1	1	1	0	0
V_6	1	0	0	1	0	0



3. 假设一个有向图具有 n 个顶点 e 条边，要求删除与顶点 i 相关联的所有边，如果该有向图采用邻接矩阵存储，删除操作的时间复杂度是多少？如果该有向图采用邻接表存储，删除操作的时间复杂度是多少？请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示，请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列



第 4 题图

3. 采用邻接矩阵存储，删除操作的时间复杂度是O(n)，采用邻接表存储的时间复杂度是O(e)，分析：邻接矩阵删除与顶点i相关联的所有边，需将矩阵第i行和第i列的所有元素置为0，时间复杂度为O(n)，邻接表删除与顶点i相关联的所有边时，需删除该顶点i的出边表中的所有结点时间复杂度为O(e)。

4. 深度优先遍历：$v_0 v_1 v_2 v_4 v_3 v_5$

广度优先遍历序列：$v_0 v_1 v_2 v_3 v_4 v_5$

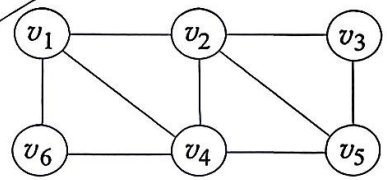
到的不同遍历序列个数是(　　)。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题(每小题 10 分,共 40 分)

1. 设无向网图 G 含有 n 个顶点 e 条边,每个顶点的信息(假设只存储编号)占用 2 字节,每条边的权值信息占用 4 字节,每个指针占用 4 字节,计算采用邻接矩阵和邻接表分别占用多少存储空间?

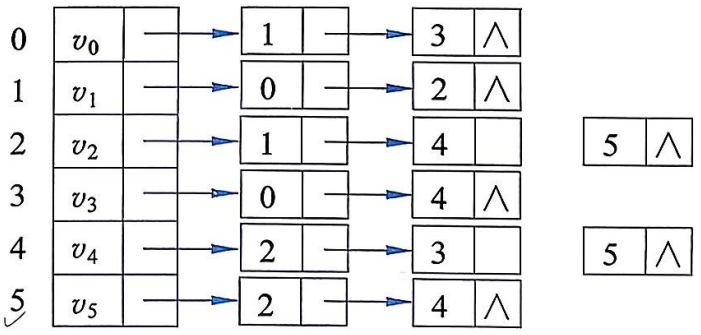
2. 已知一个连通图如下图所示,试给出图的邻接矩阵和邻接表存储示意图。



第 2 题图

3. 假设一个有向图具有 n 个顶点 e 条边,要求删除与顶点 i 相关联的所有边,如果该有向图采用邻接矩阵存储,删除操作的时间复杂度是多少?如果该有向图采用邻接表存储,删除操作的时间复杂度是多少?请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示,请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列。



第 4 题图

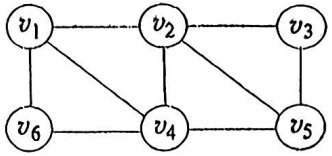
到的不同遍历序列个数是(　　)。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题(每小题 10 分,共 40 分)

1. 设无向网图 G 含有 n 个顶点 e 条边,每个顶点的信息(假设只存储编号)占用 2 字节,每条边的权值信息占用 4 字节,每个指针占用 4 字节,计算采用邻接矩阵和邻接表分别占用多少存储空间?

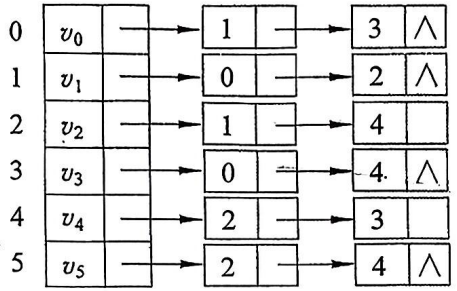
2. 已知一个连通图如下图所示,试给出图的邻接矩阵和邻接表存储示意图。



第 2 题图

3. 假设一个有向图具有 n 个顶点 e 条边,要求删除与顶点 i 相关联的所有边,如果该有向图采用邻接矩阵存储,删除操作的时间复杂度是多少?如果该有向图采用邻接表存储,删除操作的时间复杂度是多少?请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示,请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列。



第 4 题图

三、算法设计(每小题 15 分,共 30 分)

到的不同遍历序列个数是（D）。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题（每小题 10 分，共 40 分）

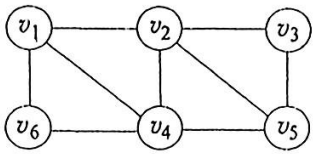
1. 设无向网图 G 含有 n 个顶点 e 条边，每个顶点的信息（假设只存储编号）占用 2 字节，每条边的权值信息占用 4 字节，每个指针占用 4 字节，计算采用邻接矩阵和邻接表分别占用多少存储空间？邻接矩阵：$4n^2+2n$　　邻接表：$2n+8e$

2. 已知一个连通图如下图所示，试给出图的邻接矩阵和邻接表存储示意图。

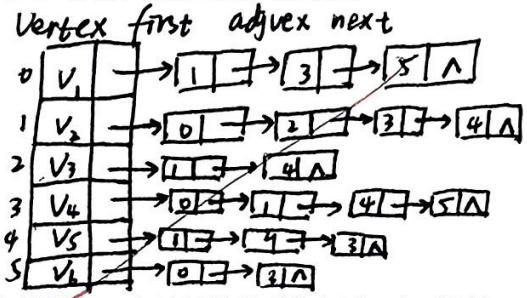
Vertex[6]: | V_1 | V_2 | V_3 | V_6 | V_4 | V_5 |

edge[6][6]:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$



第 2 题图



3. 假设一个有向图具有 n 个顶点 e 条边，要求删除与顶点 i 相关联的所有边，如果该有向图采用邻接矩阵存储，删除操作的时间复杂度是多少？如果该有向图采用邻接表存储，删除操作的时间复杂度是多少？请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示，请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列。

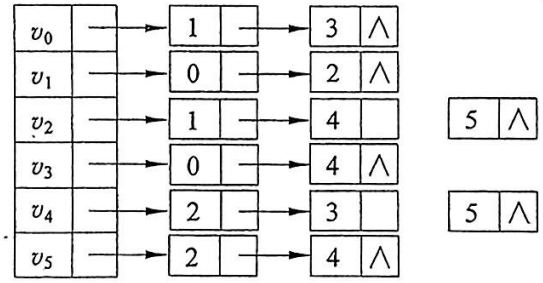
4. 深：V_0, V_1, V_2, V_4, V_5

广：V_0, V_1, V_2, V_3, V_4

V_5

3. 邻接矩阵：存在 n×n 的矩阵中，n 是顶点数量，每个元素 A[i][j] 从顶点 i 到顶点 j 的处的存在性。时间复杂度 O(n)

邻接表：链表中的元素数量等于从顶点 i 出发的处的数量，时间复杂度为 O(e)



第 4 题图

三、算法设计（每小题 15 分，共 30 分）

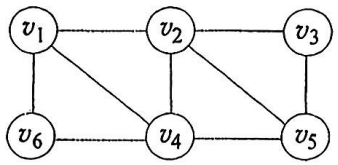
到的不同遍历序列个数是(　　)。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题(每小题 10 分,共 40 分)

1. 设无向网图 G 含有 n 个顶点 e 条边,每个顶点的信息(假设只存储编号)占用 2 字节,每条边的权值信息占用 4 字节,每个指针占用 4 字节,计算采用邻接矩阵和邻接表分别占用多少存储空间?

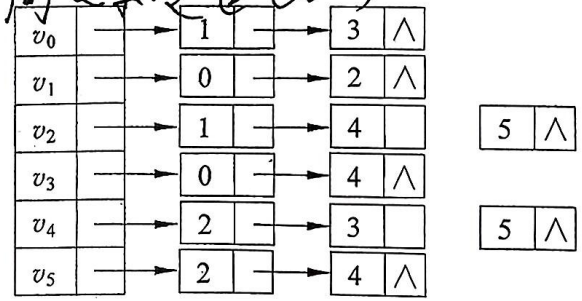
2. 已知一个连通图如下图所示,试给出图的邻接矩阵和邻接表存储示意图。



第 2 题图

3. 假设一个有向图具有 n 个顶点 e 条边,要求删除与顶点 i 相关联的所有边,如果该有向图采用邻接矩阵存储,删除操作的时间复杂度是多少?如果该有向图采用邻接表存储,删除操作的时间复杂度是多少?请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示,请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列。



第 4 题图

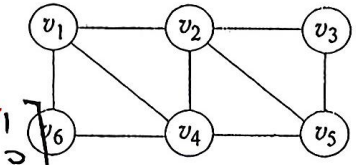
到的不同遍历序列个数是(　D　)。

A. 2　　B. 3　　C. 4　　D. 5

二、回答下列问题(每小题 10 分,共 40 分)

1. 设无向网图 G 含有 n 个顶点 e 条边,每个顶点的信息(假设只存储编号)占用 2 字节,每条边的权值信息占用 4 字节,每个指针占用 4 字节,计算采用邻接矩阵和邻接表分别占用多少存储空间?

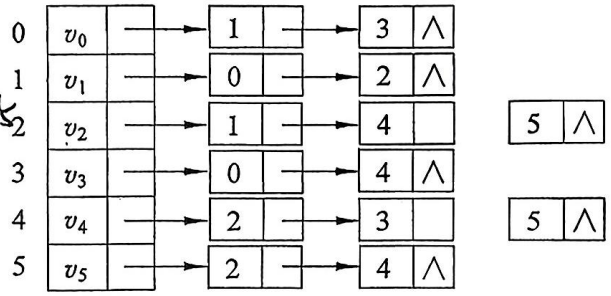
2. 已知一个连通图如下图所示,试给出图的邻接矩阵和邻接表存储示意图。



第 2 题图

3. 假设一个有向图具有 n 个顶点 e 条边,要求删除与顶点 i 相关联的所有边,如果该有向图采用邻接矩阵存储,删除操作的时间复杂度是多少?如果该有向图采用邻接表存储,删除操作的时间复杂度是多少?请给出简要分析过程。

4. 已知无向图 G 的邻接表如下图所示,请写出从顶点 v_0 出发的深度优先序列遍历和广度优先遍历序列。

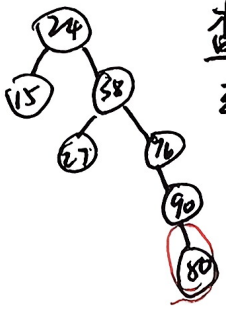


第 4 题图

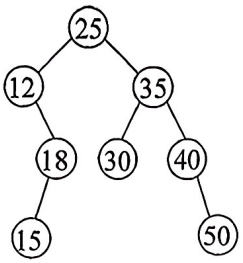
2. 对于查找集合{24，15，38，27，76，90，80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素 25 后的二叉查找树。

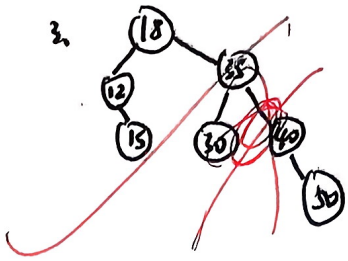
2、



查找成功的平均查找长度 2.75



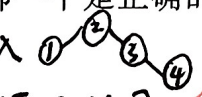
3.



第 3 题图

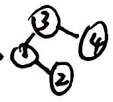
4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。 1正确

若删除4后再插入



Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同； T3≠T1 不相同

Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同； 删除2后再插入



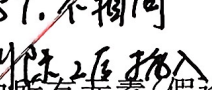
T3≠T1 不相同

Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

三、算法设计(每小题 15 分，共 30 分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

删除2后再插入



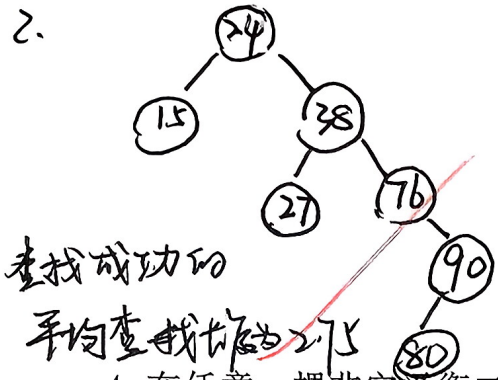
T3≠T1 不相同

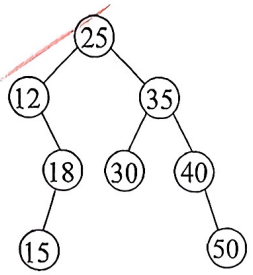
2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

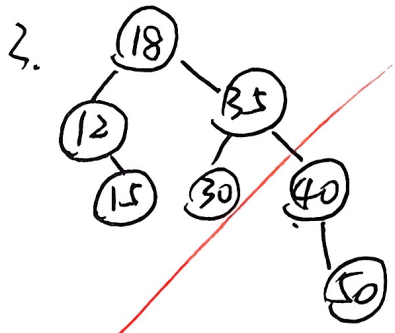
```
1. void search (int r[], int n, int x, int y)
   { int i = BinSearch (r, n, x);
     int j = BinSearch (r, n, y);
     while (i<=j)
     printf ("%d", r[i++]);
   }

2. int level (BiNode *root, BiNode *x)
   { if (root==NULL) return 0;
     if (x==root) return 1;
     if (x->date < root->date) return level (root->lchild, p) +1;
     else return level (root->rchild, p)+1;
   }
```

2. 对于查找集合{24，15，38，27，76，90，80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素25后的二叉查找树。







第3题图

4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。

Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同；

Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同；

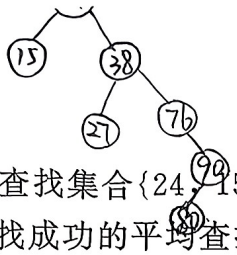
Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

三、算法设计(每小题15分，共30分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

```
(1) void Search(int r[], int n, int x, int y)
{ int i = BinSearch(r, n, x);
  int j = BinSearch(r, n, y);
    while (i<=j)
     print("%2d", r[i++]);
  }
(2) int level(BiNode *root, BiNode *X)
{ if (root == NULL) return 0;
  if (X == root) return 1;
  if (X->date < root->date) return level(root->lchild, p)+1;
  else return level(root->rchild, p)+1;
}
```

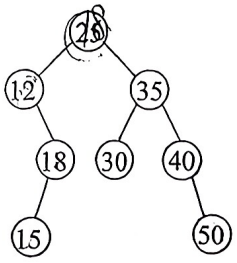


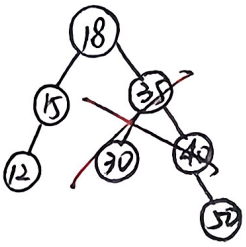
$ASL = \frac{1}{7} \times (1\times1 + 2\times2 + 3\times2 + 4\times1 + 5\times1)$

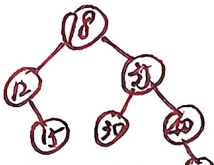
$= \frac{20}{7}$

2. 对于查找集合{24，15，38，27，76，90，80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素 25 后的二叉查找树。







第 3 题图

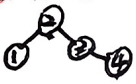
4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。仅Ⅰ正确

Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同；

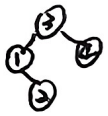
Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同；

Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

若删除4后再插入 T_3 与 T_1 不相同



删除2后插入 T_3 与 T_1 不相同



删除2后插入 T_3 与 T_1 相同



三、算法设计(每小题 15 分，共 30 分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

```
1. void Search(int r[], int n, int x, int y)
  {int i = BinSearch(r,n,x);
   int j = BinSearch(r,n,y);
   while (i≤j)·
    print("%2d", r[i++]);
  }
```

```
2. int Level(BiNode *root, BiNode *x)
  {if (root==NULL)
    return 0;
   if (x==root)
   return 1;
   if (x->data < root->data)
   return Level(root->lchild,p)+1.
   else
   return Level(root->rchild,p)+1;
   }
```

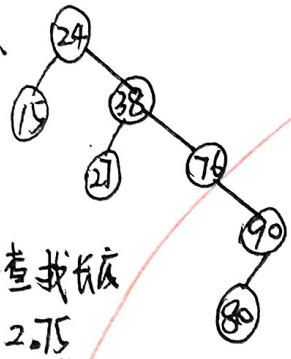
B

对于查找集合{53,17,12,61,89,70,87,25,64,46}
设散列表长度为15，……（相册）

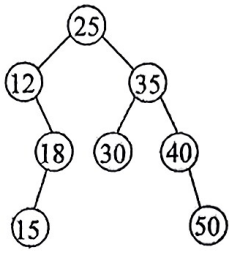
2. 对于查找集合{24, 15, 38, 27, 76, 90, 80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素 25 后的二叉查找树。

2、

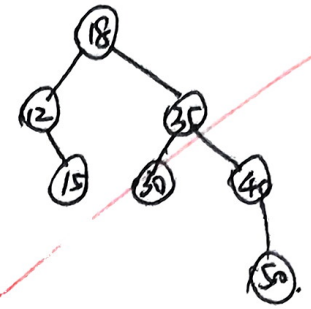


平均查找长度
2.75



第 3 题图

3、



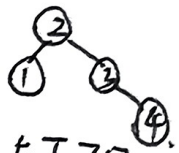
4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。 I 正确

Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同；

Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同；

Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

若删除4后再插入. T3与T1不同



删除2后插入 T3与T1不同.

删除2后，T3与T1相同

三、算法设计(每小题 15 分，共 30 分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

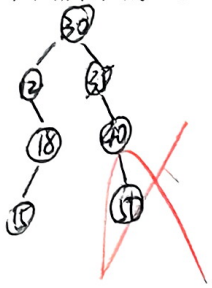
(1)
```
void Search (int r[], int n, int x, int y)
{ int i = BinSearch (r,n,x);
  int j = BinSearch (r,n,y);
  while (i<=j)
  print ("%2d",r[i++]);
}
```

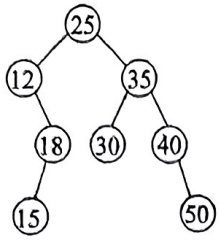
(2).
```
int level (BiNode *root, BiNode *x).
{if (root == NULL) return 0;
if (x==root) return 1;
if (x->date<root->date) return level (root->child,p)+1;
else return level root->rchild,p)+1;
}
```

B

2. 对于查找集合{24，15，38，27，76，90，80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素 25 后的二叉查找树。





第 3 题图

4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。

Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同；

Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同；

Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

三、算法设计(每小题 15 分，共 30 分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

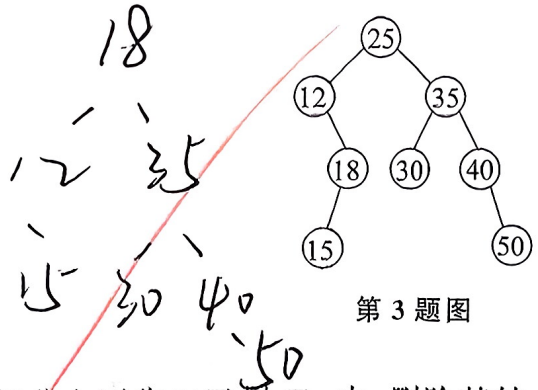
2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

```
def find_elements_in_range(r,x,y):
    left = binary_search_left(r,x)
    right = binary_search_right(r,y)
    return r[left:right]
def binary_search_left(r,x):
    left,right = 0,len(r)-1
    while left <= right:
        mid = (left+right)//2
        if r[mid]<x:
            left = mid+1
        else:
            right = mid-1
    return left
def binary_search_right(r,y):
    left,right = 0,len(r)-1
    while left <= right:
        mid = (left+right)//2
        if r[mid] <= y:
            left = mid+1
        else:
            right = mid-1
    return left
```

```
function findLevel(root,x):
    current = root
    level = 1
    while current is not null:
        if current.value == x:
            return level
        elif x < current.value:
            current = current.left
        else:
            current = current.right
        level += 1
    return -1
```

2. 对于查找集合{24，15，38，27，76，90，80}，请画出构建的二叉查找树，并求等概率情况下查找成功的平均查找长度。

3. 对下图所示的二叉查找树，画出删除元素 25 后的二叉查找树。



第 3 题图

4. 在任意一棵非空平衡二叉树 T_1 中，删除某结点 v 之后形成平衡二叉树 T_2，再将 v 插入 T_2 形成平衡二叉树 T_3。下列关于 T_1 与 T_3 的叙述中，哪一个是正确的？请给出分析过程。

Ⅰ. 若 v 是 T_1 的叶子结点，则 T_1 与 T_3 可能不相同；

Ⅱ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定不相同；

Ⅲ. 若 v 不是 T_1 的叶子结点，则 T_1 与 T_3 一定相同。

三、算法设计(每小题 15 分，共 30 分)

1. 设数组 r[n]为升序序列，请查找元素值在 x 和 y 之间的所有元素(假设 $x<y$)。

2. 给定元素值 x，求结点 x 在二叉查找树中的层数。

```
def find_elements_in_range(r, x, y)
left = binary_search_left(r, x)
right = binary_search_right(r, y)
return r[left:right]
def binary_search_left(r, x)
left, right = 0, len(r)-1
while left <= right:
mid = (left + right) //2
if r[mid] < x:
left = mid+1
else:
right = mid-1
return left
```

```
def binary_search_right(r,y)
left, right = 0, len(r)-1
while left <= right:
mid = (left+right) //2
if r[mid] <= y
left = mid+1
else right = mid-1
return left
```

1. ①散列函数是均匀 ②处理冲突的方法 ③散列表的装填因子

2. 散列地址

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
关键码		53			17	70		46	64	61	87	29	12	25	
比较次数		1			1	1		1	5	1	2	1	5	2	

$\frac{(1\times7+2\times2+5\times3)}{10}=1.6$

16. 设主串 S="*abccdcdccbaa*"，模式串 T="*cdcc*"，采用 KMP 算法进行模式匹配，则在第（ B ）趟匹配成功。

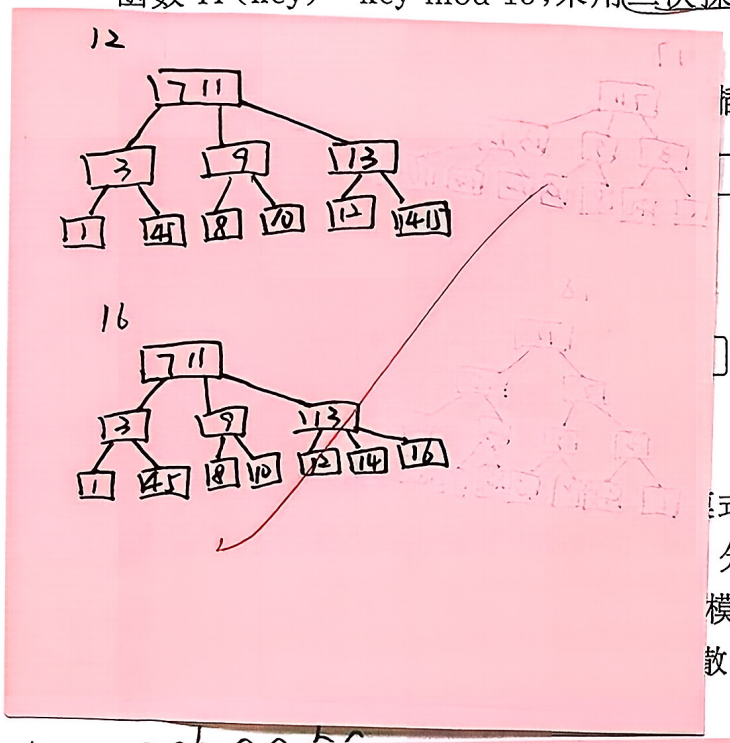
A. 1　　B. 3　　C. 5　　D. 7

64
13 | 53
5 82

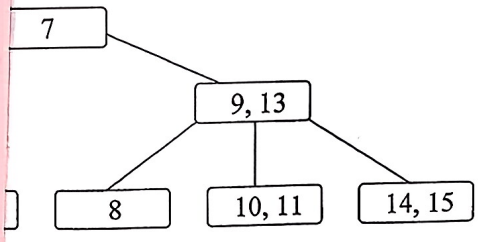
二、回答下列问题（每小题 10 分，共 40 分）

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成



插入关键码 12，16，17 和 18 之后的结果。



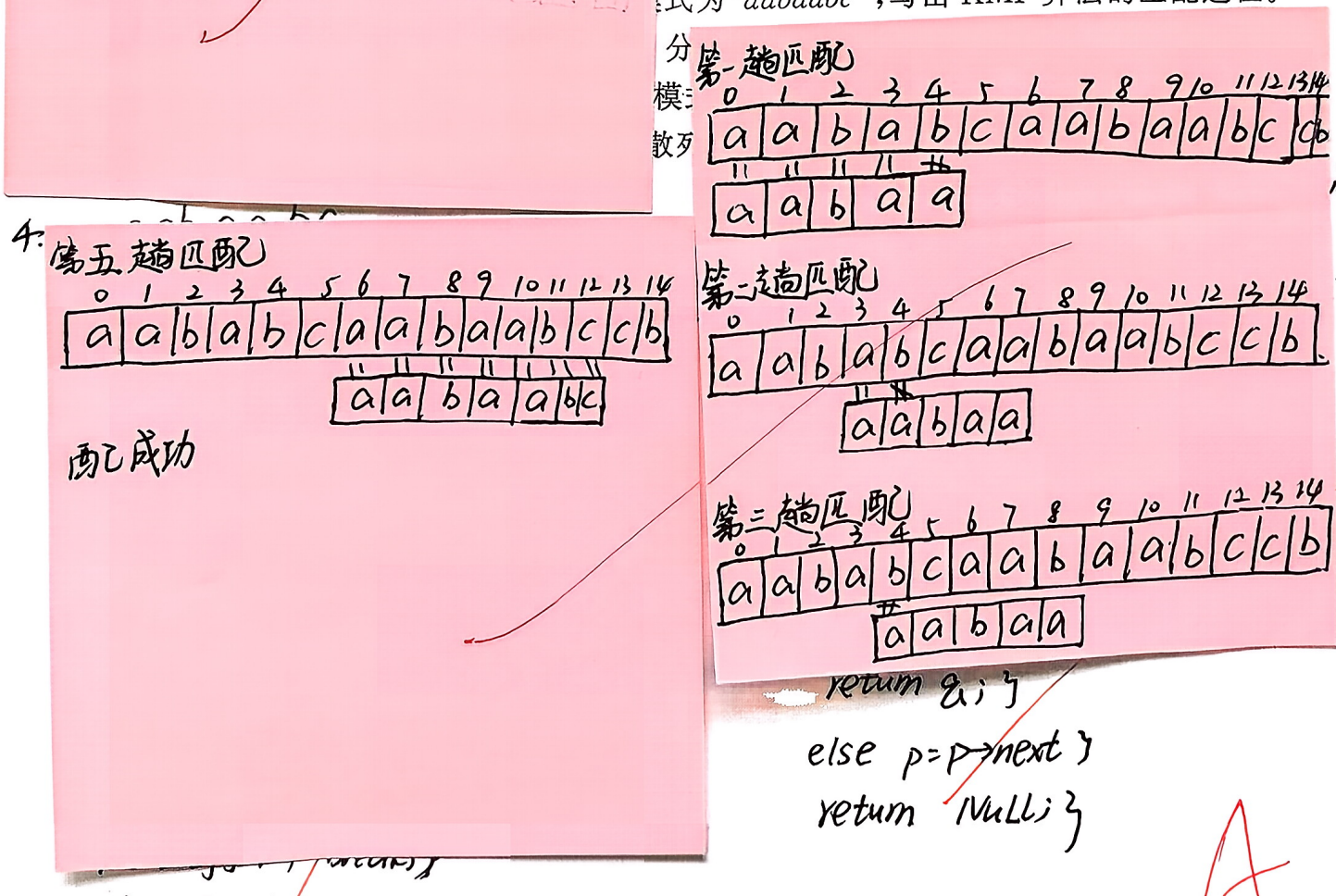
第 3 题图

式为"*aabaabc*"，写出 KMP 算法的匹配过程。

分

模

散列



int m;

*=NULL;

return q;}

else p=p->next;}

return NULL;}

else k--;}

if (0==k)

next[j]=0;

10. 设主串 S="*abccdcdccbaa*"，模式串 T="*cdcc*"，采用 KMP 算法进行模式匹配，则在第（　）趟匹配成功。

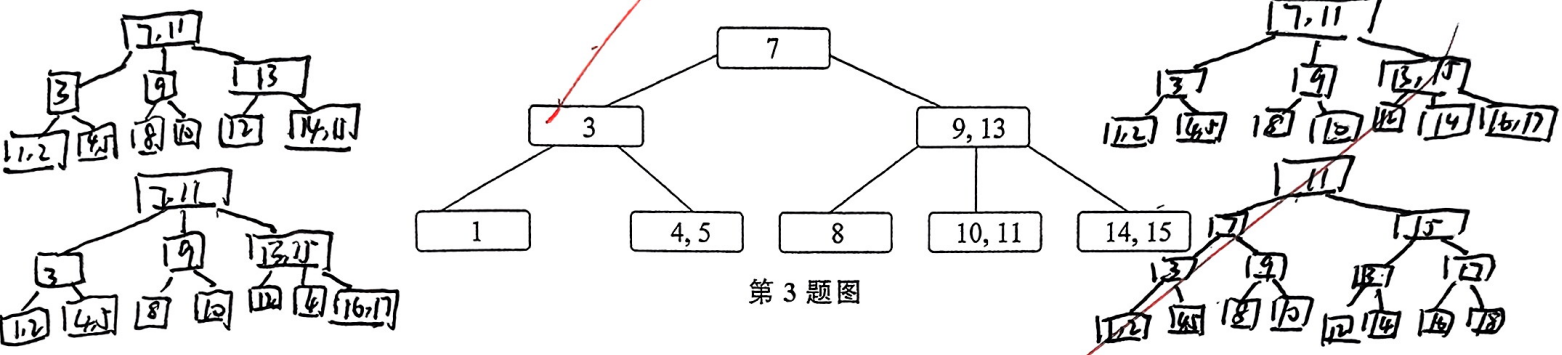
A. 1　　B. 3　　C. 5　　D. 7

二、回答下列问题（每小题 10 分，共 40 分）

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成功的平均查找长度。

3. 对下图所示 3 阶 B 树，分别给出插入关键码 12，16，17 和 18 之后的结果。



第 3 题图

4. 设主串为"*aabaabcaabaabccb*"，模式为"*aabaabc*"，写出 KMP 算法的匹配过程。

三、算法设计（每小题 15 分，共 30 分）

1. 对于 KMP 算法，请设计函数求模式 $T=t_1t_2\cdots t_m$ 的匹配失效位置 next[m]。

2. 在用拉链法解决冲突构造的开散列表 ht[n]中，删除值为 x 的记录。

```
#include<stdio.h>
#include<string.h>
void computeNext(const char *T, int next[]){
    int m = strlen(T);
    next[0]=-1;
    int i=0, j=-1;
    while(i<m){
        if(j==-1||T[i]==T[j]){
            i++;
            j++;
            next[i]=j;
        }else{
            j=next[j];
        }
    }
}
int main(){
    char pattern[]="ababaca";
    int m=strlen(pattern);
    computeNext(pattern,next);
    printf("Pattern: %s\n",pattern);
    printf("Next array: ");
    for(int i=0;i<=m;i++){
        printf("%d ",next[i]);
    }
    printf("\n");
    return 0;
}
```

10. 设主串 S="*abccdcdccbaa*"，模式串 T="*cdcc*"，采用 KMP 算法进行模式匹配，则在第（　　）趟匹配成功。

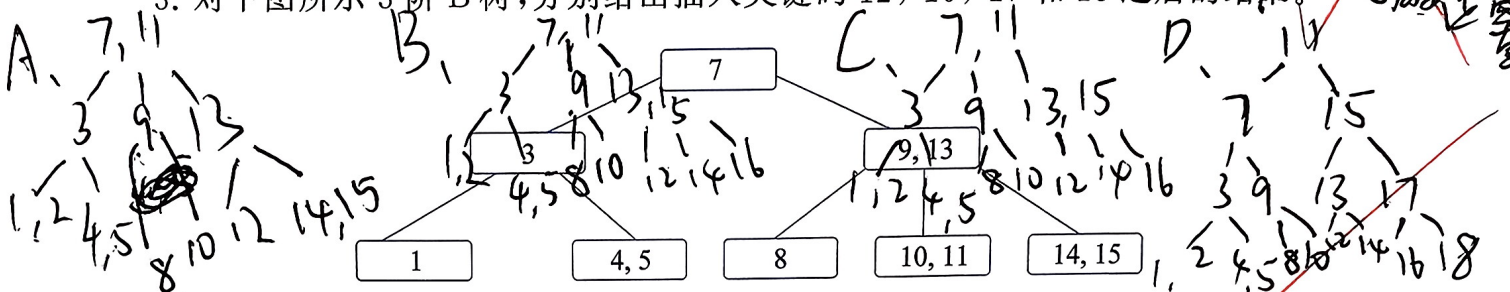
A. 1　　B. 3　　C. 5　　D. 7

二、回答下列问题（每小题 10 分，共 40 分）

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成功的平均查找长度。

3. 对下图所示 3 阶 B 树，分别给出插入关键码 12，16，17 和 18 之后的结果。



第 3 题图

4. 设主串为"*aababcaabaabccb*"，模式为"*aabaabc*"，写出 KMP 算法的匹配过程。

三、算法设计（每小题 15 分，共 30 分）

1. 对于 KMP 算法，请设计函数求模式 T="$t_1t_2\cdots t_m$"的匹配失效位置 next[m]。

2. 在用拉链法解决冲突构造的开散列表 ht[n]中，删除值为 x 的记录。

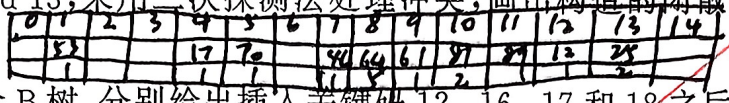
10. 设主串 S="*abccdcdccbaa*"，模式串 T="*cdcc*"，采用 KMP 算法进行模式匹配，则在第（B）趟匹配成功。

A. 1　　B. 3　　C. 5　　D. 7

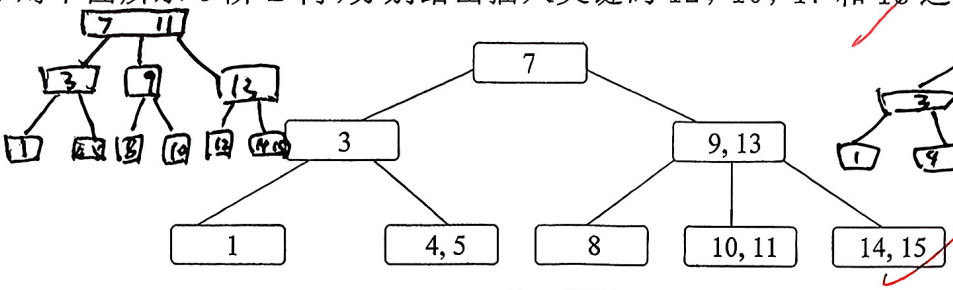
二、回答下列问题（每小题 10 分，共 40 分）

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成功的平均查找长度。



3. 对下图所示 3 阶 B 树，分别给出插入关键码 12，16，17 和 18 之后的结果。



第 3 题图

4. 设主串为"*aababcaabaabccb*"，模式为"*aabaabc*"，写出 KMP 算法的匹配过程。

三、算法设计（每小题 15 分，共 30 分）

1. 对于 KMP 算法，请设计函数求模式 T="$t_1t_2\cdots t_m$"的匹配失效位置 next[m]。

2. 在用拉链法解决冲突构造的开散列表 ht[n]中，删除值为 x 的记录。

```
1. void Next (char t[], int next[], int m);
{ int i, k;
  next[0] = -1;
  next[1] = 0;
  for (i=2; i<=m; j++)
  { k = j-1;
    while (k>0)
    { for (i=0; i<k; i++)
      if (t[i] != t[j-k+i]) break;
      if (i==k)
      { next[j] = k; break; }
      else k--; }
    if (k==0) next[j] = 0;
  }
}
```

```
2. Node *HashDelete (Node *ht[],
int m, int x)
{ int j = H(x); Node *p = ht[j], *q =
  NULL; if (p->data == x)
{ ht[j] = p->next; return p; }
while (p->next != NULL)
{ if (p->next->data == x)
{ q = p->next;
  p->next = q->next;
  return q; }
  else p = p->next;
  return NULL;
}
```

10. 设主串 S="*abccdcdccbaa*"，模式串 T="*cdcc*"，采用 KMP 算法进行模式匹配，则在第（C）趟匹配成功。

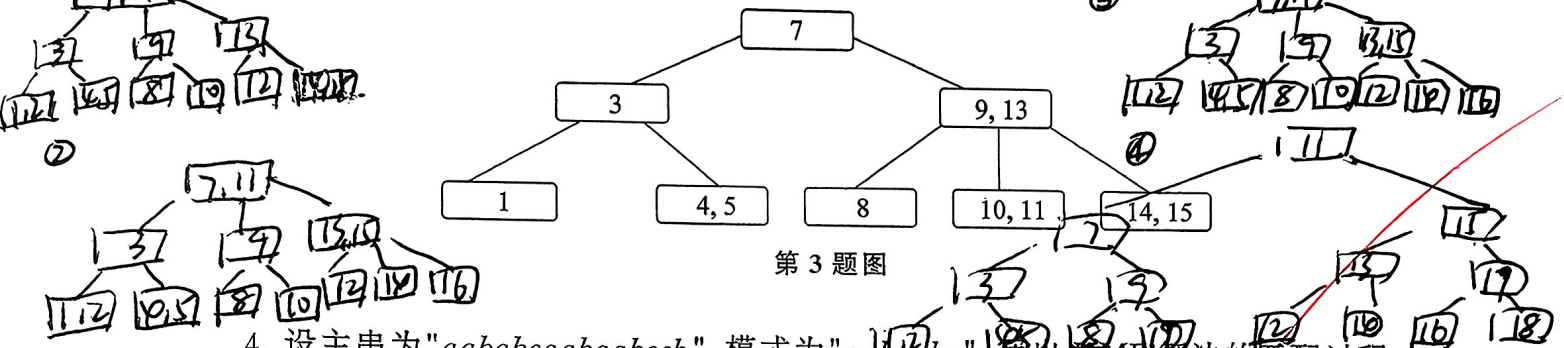
A. 1　　B. 3　　C. 5　　D. 7

二、回答下列问题（每小题 10 分，共 40 分）

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成功的平均查找长度。

3. 对下图所示 3 阶 B 树，分别给出插入关键码 12，16，17 和 18 之后的结果。



第 3 题图

4. 设主串为"*aababcaabaabccb*"，模式为"*aabaabc*"，写出 KMP 算法的匹配过程。

三、算法设计（每小题 15 分，共 30 分）

1. 对于 KMP 算法，请设计函数求模式 T="$t_1t_2\cdots t_m$"的匹配失效位置 next[m]。

2. 在用拉链法解决冲突构造的开散列表 ht[n]中，删除值为 x 的记录。

10. 设主串 S="$abccdcdccbaa$"，模式串 T="$cdcc$"，采用 KMP 算法进行模式匹配，则在第（ C ）趟匹配成功。

A. 1　　B. 3　　C. 5　　D. 7

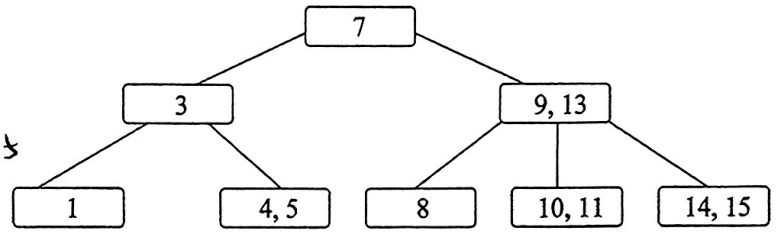
二、回答下列问题（每小题 10 分，共 40 分）

黑载图，哈希函数的质量和冲突解决策略

1. 对具有 n 个关键码的散列表进行查找，请分析决定平均查找长度的因素。

2. 对于查找集合{53，17，12，61，89，70，87，25，64，46}，设散列表长为 15，散列函数 H(key)＝key mod 13，采用二次探测法处理冲突，画出构造的闭散列表，求查找成功的平均查找长度。

3. 对下图所示 3 阶 B 树，分别给出插入关键码 12，16，17 和 18 之后的结果。



第 3 题图

aababcaabaabccb
aabaa
aa
a
a
aabaaabc

4. 设主串为"$aababcaabaabccb$"，模式为"$aabaabc$"，写出 KMP 算法的匹配过程。

三、算法设计（每小题 15 分，共 30 分）

1. 对于 KMP 算法，请设计函数求模式 T="$t_1t_2\cdots t_m$"的匹配失效位置 next[m]。

2. 在用拉链法解决冲突构造的开散列表 ht[n]中，删除值为 x 的记录。

```
1. void getNext(char T[], int next[], int m)
   {next[1]=0;
   int i=0;
   int j=2;
   while(j<=m)
   {if(i==0 || T[i]=T[j-1])
   {i++; next[j]=i; j++;}
   else {i=next[i]; j}}
```

```
Node *prev*NULL;
Node *curr=ht->table[index];
while(curr!=NULL && curr->
        data!=key)
{prev=curr;
 curr=curr->next;}
if(curr=NULL)
{return ;}
if(prev==NULL)
{ht->table[index]=curr->next
else
prev->next=
curr->next;}
```

```
2. # include <stadio.h>
   # include <stdib.h>
   # include <string.h>
   # define TABLE_SIZE 10
   typedef struct Node;
   int data;
   struct Node *next; } Node;
   {Node * table [TABle_SIZE]; }
   HaSh Table;
```

```
void initHash Table(Hash Table * ht)
{for( int i=0; i<TABLE_SIZE; i++)
 {for(int i=0; i<TABLE_SIZE; i++)
   {ht->Table[i] NULL ;}
   int hash Fuction(int key)
   {return key % TABLE ;}
void insert(Hash Table *ht, int key)
{int index=hash Fuction(key);
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

堆满足性质,用于实现优先队列
二叉查找树用于快速查找、插入和删除

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

排序方法,时间复杂度为O(nlogn),在这些排序方法中,在最坏的时候有较好的性能

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

最好:1,2,3,4,5,6,7,8 比较最小值 17
最坏:8,7,6,5,4,3,2,1 大 28

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;(H,C,Q,P,A,M,S,R,D,F,X,Y)

(2) 增量为 4 的希尔排序一趟扫描的结果;(P,A,C,S,Q,D,F,X,R,H,M,Y)

(3) 二路归并排序第一趟扫描的结果;(H,Q,C,Y,A,P,M,S,D,R,F,X)

(4) 以第一个元素为轴值的快速排序一趟排序的结果;(F,H,C,D,P,A,M,Q,R,S,Y,X)

(5) 堆排序初始建堆的结果。(A,D,C,R,F,Q,M,S,Y,P,H,X)

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
1. void StraightSort(Node *first)
{
  pre=first; p=first->next; q=p->next;
  while(q!=NULL)
  {
    while(p->data<q->data)
    {
      pre=p; p=p->next;
    }
    if(p!=q){
      u=q->next; pre->next=q;
      q->next=p; q=u;
    }
    else q=q->next;
  }
  pre=first; p=first->next;
}
```

```
2. void de_SiftHeap(int r[], int k, int n)
{
  int i,j,temp,temp1;
  i=k; j=2*i+1;
  if(i<0||i>n-1)
  return error;
  else if(i==n-1)
  free(a[i]);
  else
  while(j<n)
  {
  temp1=a[i];
  a[i]=a[n-1];
  a[n-1]=temp1;
  if(j<n-1&&r[j]<r[j+1])j++;
  if(r[i]>r[j])
    break;
  else{
    temp=r[i]; r[i]=r[j];
  r[j]=temp;
    i=j; j=2*i+1;
  }
  }
}
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

堆满足性质,用于实现优先队列
二叉查找树用于快速查找、插入和删除

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

排序方法,时间复杂度为O(nlogn),在这些排序方法中,在最坏的时候有较好的性能

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

最好:1,2,3,4,5,6,7,8 比较最小值 1
最坏:8,7,6,5,4,3,2,1 大 28

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;(H,C,Q,P,A,M,S,R,D,F,X,Y)

(2) 增量为 4 的希尔排序一趟扫描的结果;(P,A,C,S,Q,D,F,X,R,H,M,Y)

(3) 二路归并排序第一趟扫描的结果;(H,Q,C,Y,A,P,M,S,D,R,F,X)

(4) 以第一个元素为轴值的快速排序一趟排序的结果;(F,H,C,D,P,A,M,Q,R,S,Y,X)

(5) 堆排序初始建堆的结果。(A,D,C,R,F,Q,M,S,Y,P,H,X)

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
1. void StraightSort(Node *first)
{
  pre=first; p=first->next; q=p->next;
  while(q!=NULL)
  {
    while(p->data<q->data)
    {
      pre=p; p=p->next;
    }
    if(p!=q){
      u=q->next; pre->next=q;
      q->next=p; q=u;
    }
    else q=q->next;
  }
  pre=first; p=first->next;
 }
}
```

```
2. void de_SiftHeap(int r[], int k, int n)
{
  int i,j,temp,temp1;
  i=k; j=2*i+1;
  if(i<0||i>n-1)
  return error;
  else if(i==n-1)
  free(a[i]);
  else
  while(j<n)
  {
  temp1=a[i];
  a[i]=a[n-1];
  a[n-1]=temp1;
  if(j<n-1&&r[j]<r[j+1])j++;
  if(r[i]>r[j])
    break;
  else{
    temp=r[i]; r[i]=r[j];
  r[j]=temp;
    i=j; j=2*i+1;
  }
  }
}
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

4. 设待排序记录序列是$(Q, H, C, Y, P, A, M, S, R, D, F, X)$,请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果;

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟排序的结果;

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
1. void StraightSort(Node* first)
{ pre=first; p=first->next; q=p->next;
  while(q!=NULL)
{ while(q!=p;
  { while(p->data<q->data)
    { pre=p; p=p->next; }
    if(p!=q){ u=q->next;
      pre->next=q; q->next=p; q=u; }
    else q=q->next;
    pre=first; p=first->next; } }
```

```
2. void adjust-heap(int* a,
   int node, int size)
{ int left=i*node;
  if(left<size && a[left]>a[max]
  max=left;
  if(right<size && a[right]>a[max]
  max=right;
  if(max!=node)
  { swap(a[max], a[max])
    交换
    adjust-heap(a, max,
      size);
  }
}
```

B

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结

2. 在大根堆中删除最大值结点,要求算法

解: (1) (H, C, Q, P, A, M, S, R, D, F, X

(2) (P, A, C, S, Q, D, F, X, R, H,

(5) (A, D, C, R, F, Q, M, S, Y, P, H

二.1. 二叉查找树是结点之间满足一定次序关系的二叉树。堆是结点之间满足一定次序关系的完全二叉树.

2. 基数排序, 英语单词比较长, 设n个单词中最长单词有m个字母, 时间复杂度为 O(m(n+26m)), 即 O(n)

3. 最好: 7+2+3+1=13

最坏: 1+2+3+4+5+6+7+8=28

4. (3) (H, Q, C, Y, A, P, M, S, D, R, F, X)

(4) (F, H, C, D, P, A, M, Q, R, S, Y, X)

```
1. Node *StraightSort (Node *first)
{ Node *q=NULL, *pre=NULL, *p=NULL, *u=NULL)
  q=first->next->next;
  first->next->next=NULL;
  while (q!=NULL)
  { pre=first; p=first->next;
  while (p!=NULL && p->data<q->data)
   {pre=p, p=p->next;}
    u=q->next
    pre->next=q;
    q->next=p;
    q=u;}
    return first;
   }
```

```
{int x=r[1],  r[1]=r[n];
 int temp, i=1, j=2*i;
 while (j<n)
{if (j<n-1 && r[j]>r[j])
  j++;
 if (r[i]>r[j])
 break;
 temp=r[j];
 r[j]=r[i];
 r[i]=temp;
  i=j;
 j=2*i
 return x;}
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果;

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟排序的结果;

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
1. void straightSort(Node *first)
{pre=first, p=first->next; q=p->next;
while (q!=NULL)
 {while (q!=p)
 {while (p->data<=q->data)
  {pre=p; p=p->next;}
 if (p!=q)
 {u=q->next; pre->next=q;
 q->next=p; q=u;}
 else q=q->next;}
 pre=first; p=first->next;}}
```

```
2. void de_SiftHeap(int r[], int k, int n)
{int i, temp1;
 i=k; j=2*i+1;
 if(i<0||i>n-1)
return error;
else if(i==n-1)
free(a[i]);
else
while(j<n)
{temp1=a[i];
 a[i]=a[n-1];
 a[n-1]=temp1;
if(j<n-1&&r[j]<r[j+1])
j++;
if(r[i]>r[j])
break;
else
temp=r[i]; r[i]=r[j]; r[j]=temp;
i=j; j=2*i+1;}}
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果;

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟排序的结果;

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
1. void straightSort(Node* first)
{
pre=first; p=first->next; q=p->next
while(q!=NULL)
{while(p->data<=q->data)
{pre=p; p=p->next;}
if(p!=q){
u=q->next; pre->next=q
q->next=p; q=u;}
else q=q->next;}
pre=first; p=first->next;}
```

```
2. void de_SiftHeap(int r[];
                    int n)
{int i,j,temp,temp1;
i=k; j=2*i+1
if(i<0||j>n-1)
return error.
else if(i==n-1)
free(a[i])
else
while(j<n)
{temp1=a[i];
a[i]=a[n-1]
a[n-1]=temp1;
if(j<n-1&&r[j]<r[j+1])j++
if(r[i]>r[j]
break;
else{
temp=r[i]; r[i]=r[j]; r[j]=r[i]=temp
i=j; j=2*j+1;
}}}
```

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别?

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果;

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟排序的结果;

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

1、定义 pre 指针指向链表的结点

p 指向的结点下一个结点

q 指向 p 的下一个结点

2. while (q!=null)

p->data < q->data

pre = pre++

p = p+1

D

A. k　　B. $2k$　　C. $2k+1$　　D. $2k-1$

二、回答下列问题(每小题 10 分,共 40 分)

1. 请说明堆和二叉查找树有什么区别? 二叉查找数用于快速查找插入 ~~删~~ 除

2. 将 1000 个英文单词进行排序,采用哪种排序方法时间性能最好,请说明原因。

3. 对 $n=8$,给出快速排序一个最好情况和最坏情况的初始排列的实例,说明比较次数可能达到的最大值和最小值分别是多少?

好:12348678　　坏:87654321

最小:17　　最大:28

4. 设待排序记录序列是(Q, H, C, Y, P, A, M, S, R, D, F, X),请写出如下结果。

(1) 起泡排序第一趟扫描的结果;

(2) 增量为 4 的希尔排序一趟扫描的结果;

(3) 二路归并排序第一趟扫描的结果;

(4) 以第一个元素为轴值的快速排序一趟排序的结果;

(5) 堆排序初始建堆的结果。

三、算法设计(每小题 15 分,共 30 分)

1. 设待排序的记录序列用单链表作存储结构,请设计直接插入排序算法。

2. 在大根堆中删除最大值结点,要求算法的时间复杂度为 $O(\log_2 n)$。

```
Void strighe sort(Node* first)
{pre=first, p=first->next; q=p->next;
while(q:=Null)
{while(q:=p)
 {while(p->data:q->date)
  {pre=p; p=p->next;
   }
 if(p:=q}
  u=q->next; pre->next->q;
  q->next=p; q=u;
  }
 else q=q->next;
  }
 pre=first; p=first->next;
 }
```

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-29 18:20	页面停留时长	1时 53分 10秒	26	- -	- -	98.0/100.0	优秀

阶段成绩

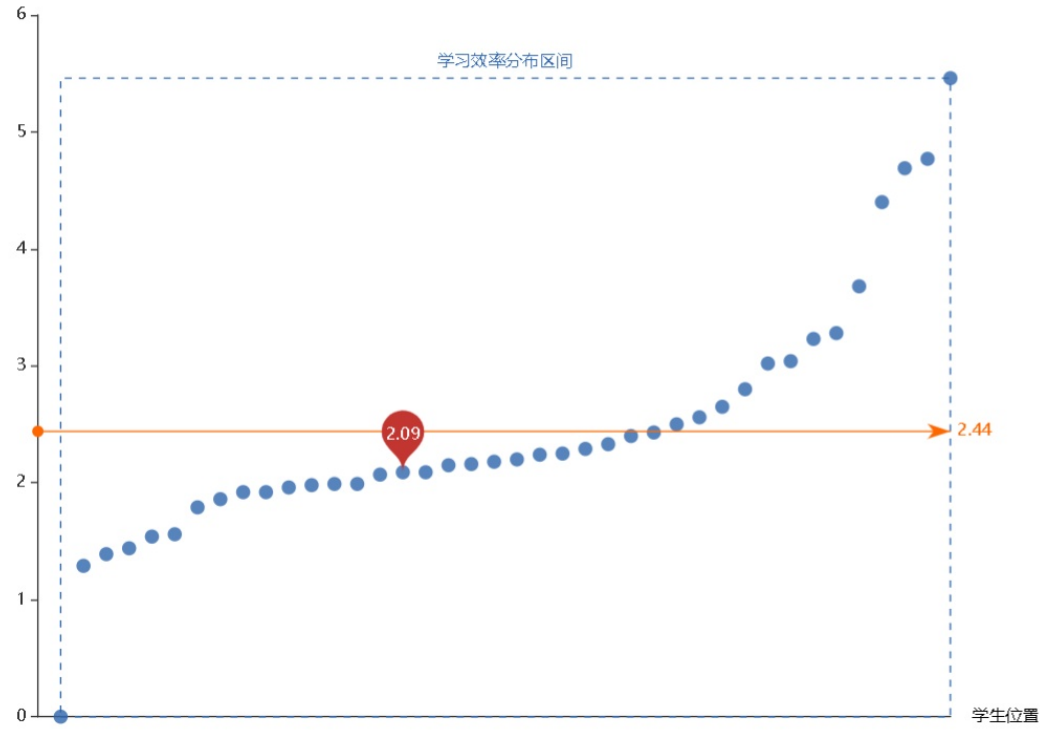
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 10:21	122	2	2024-04-23 11:36	1时 14分 26秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-04-23 11:26	117	11	2024-04-29 18:19	20分 20秒	否	100/100	15.66/16.66	15.66
3	销毁线性表	2024-04-23 11:27	125	2	2024-04-23 11:41	48秒	否	100/100	16.66/16.66	16.66
4	查找	2024-04-23 11:44	129	1	2024-04-23 11:44	5秒	否	100/100	15.66/16.66	15.66
5	删除	2024-04-23 11:44	129	1	2024-04-23 11:44	4秒	否	100/100	16.66/16.66	16.66
6	顺序表应用	2024-04-23	117	9	2024-04-29	17分 27秒	否	300/30	16.70/16.7	16.7

	11:49	18:20		0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	李雯君
学号	2340850043
学习效率	2.09
当前排名	24

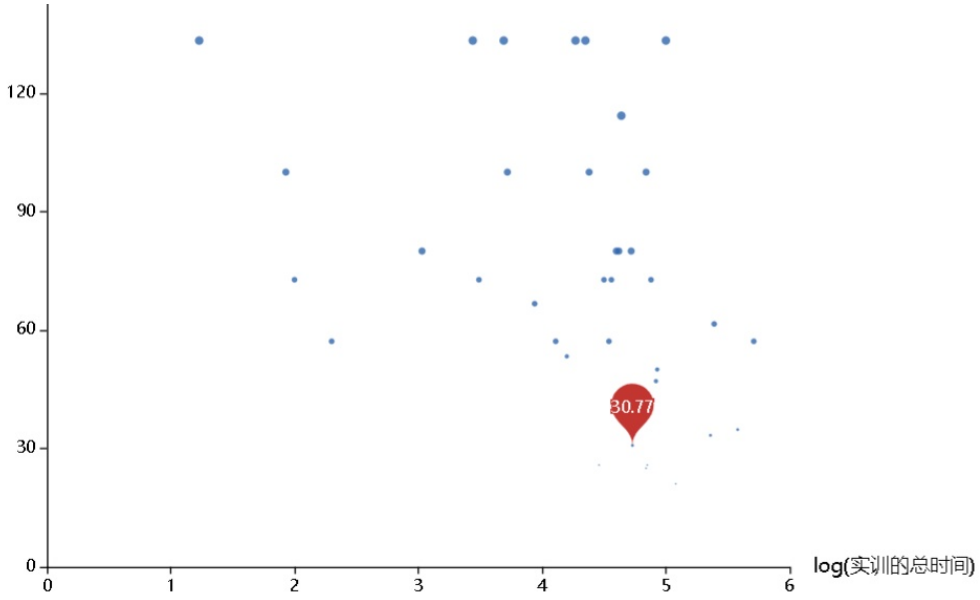
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	30.77

150



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空顺序线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 122　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	529.77	2024-04-23 11:26	查看代码
2	评测通过	0.03	1894.83	2024-04-23 11:36	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 顺序线性表插入

调分：15.66

学员代码 ⓘ　　代码修改行数 117　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.01	1894.83	2024-04-23 11:32	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.04	506.31	2024-04-23 11:42	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0.05	1887.63	2024-04-23 11:43	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0.01	574.08	2024-04-23 11:47	查看代码
5	共有3组测试集，其中有3组测试结果不匹配	0.03	1935.78	2024-04-23 11:48	查看代码
6	共有3组测试集，其中有3组测试结果不匹配	0	3659.58	2024-04-25 20:11	查看代码
7	共有3组测试集，其中有3组测试结果不匹配	0.01	138.48	2024-04-25 20:12	查看代码

8	共有3组测试集，其中有3组测试结果不匹配	0	3659.58	2024-04-25 20:13	查看代码
9	共有3组测试集，其中有3组测试结果不匹配	0.01	3500.88	2024-04-29 18:11	查看代码
10	共有3组测试集，其中有3组测试结果不匹配	0.04	3661.14	2024-04-29 18:17	查看代码

< 2 >

代码评价

一键AI生成 编辑

暂无评价~

第3关 销毁线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 125　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	486.99	2024-04-23 11:41	查看代码
2	评测通过	0.15	2156.1	2024-04-23 11:41	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找

调分：15.66

学员代码 ⓘ　　代码修改行数 129　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	324.66	2024-04-23 11:44	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 删除

调分：16.66

学员代码 代码修改行数 129 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	631.61	2024-04-23 11:44	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 顺序表应用

调分：16.7

学员代码 ⓘ　　代码修改行数 117　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	337.54	2024-04-23 11:49	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	382.72	2024-04-23 11:52	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.01	96.2	2024-04-23 11:54	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	334.08	2024-04-23 11:55	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	0	2024-04-23 11:56	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0.02	393.82	2024-04-23 11:57	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0	332.14	2024-04-25 20:08	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	378.3	2024-04-25 20:09	查看代码
9	评测通过	0.01	2439.84	2024-04-29 18:20	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



张佳慧 按时通关

学号：2340850044　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-29 18:14	页面停留时长	2时 7分 55秒	31	- -	- -	93.0/100.0	优秀

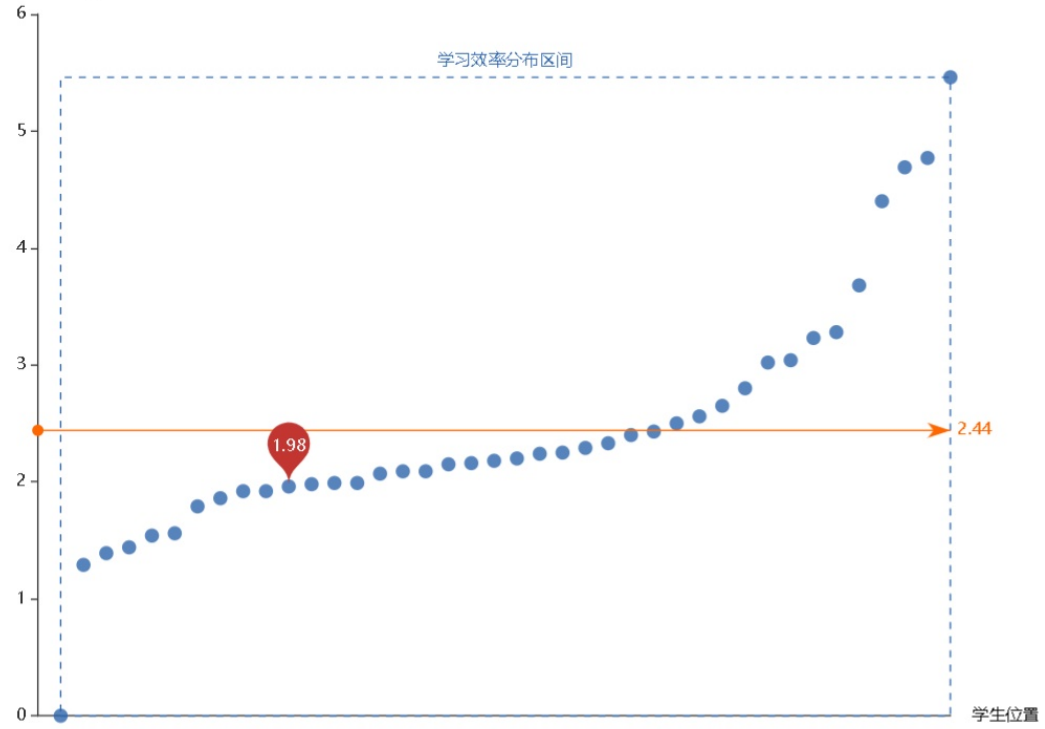
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 10:22	102	2	2024-04-23 11:21	52分 11秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-04-23 10:33	112	3	2024-04-24 21:49	13分 20秒	否	100/100	16.66/16.66	16.66
3	销毁线性表	2024-04-23 11:17	120	4	2024-04-23 11:43	12分 39秒	否	100/100	16.66/16.66	16.66
4	查找	2024-04-23 11:17	120	2	2024-04-23 11:43	10分 4秒	否	100/100	16.66/16.66	16.66
5	删除	2024-04-23 11:17	120	2	2024-04-23 11:43	37秒	否	100/100	10.66/16.66	10.66
6	顺序表应用	2024-04-23 11:18	117	18	2024-04-29 18:14	39分 4秒	否	300/300	15.70/16.7	15.7

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	张佳慧
学号	2340850044
学习效率	1.98
当前排名	29

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

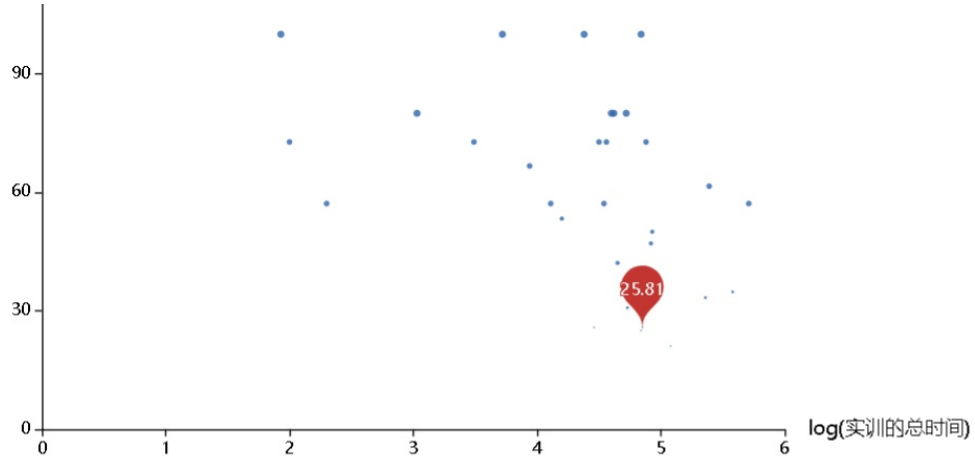
能力值

能力值（实训获得经验值/实训评测次数）



姓名	张佳慧
学号	2340850044
能力	25.81

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建空顺序线性表

调分： 16.66

学员代码 代码修改行数 102 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
  int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
  int curNum;//用于存放顺序线性表中数据元素的个数 整型 curNum
  DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	3659.82	2024-04-23 11:01	查看代码
2	评测通过	0	484.62	2024-04-23 11:21	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第2关 顺序线性表插入

调分：16.66

学员代码 ⓘ　　代码修改行数 112　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.04	3109.29	2024-04-23 11:21	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.02	139.44	2024-04-23 11:43	查看代码
3	评测通过	0.03	3193.8	2024-04-24 21:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 销毁线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 120　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有2组测试结果不匹配	0.03	2975.43	2024-04-23 11:25	查看代码
2	共有3组测试集，其中有2组测试结果不匹配	0.01	3109.29	2024-04-23 11:27	查看代码
3	共有3组测试集，其中有2组测试结果不匹配	0.03	3659.07	2024-04-23 11:36	查看代码
4	评测通过	0.03	3418.11	2024-04-23 11:43	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找

调分：16.66

学员代码 ⓘ　　代码修改行数 120　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2439.78	2024-04-23 11:25	查看代码
2	评测通过	0.03	461.4	2024-04-23 11:43	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 删除

调分：10.66

学员代码 ⓘ　　代码修改行数 120　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	1219.89	2024-04-23 11:25	查看代码
2	评测通过	0.01	46.48	2024-04-23 11:43	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第6关 顺序表应用

调分：15.7

学员代码 ⓘ　　代码修改行数 117　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
int curNum; //用于存放顺序线性表中数据元素的个数 整型 curNum
DataType* element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
//输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.03	1983.62	2024-04-23 11:25	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.03	461.4	2024-04-23 11:39	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.01	325.84	2024-04-23 11:44	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0	2129.2	2024-04-24 21:50	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	119.18	2024-04-24 21:51	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0	326.68	2024-04-24 21:52	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0	388.62	2024-04-24 21:52	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	95.24	2024-04-24 21:56	查看代码
9	共有2组测试集，其中有2组测试结果不匹配	0	323.18	2024-04-24 21:56	查看代码
10	共有2组测试集，其中有2组测试结果不匹配	0	306.64	2024-04-24 21:58	查看代码

< 2 >

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



厉耀阳 按时通关

学号：2340850052　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-23 15:09	页面停留时长	1时 33分 55秒	14	- -	- -	86.0/100.0	良好

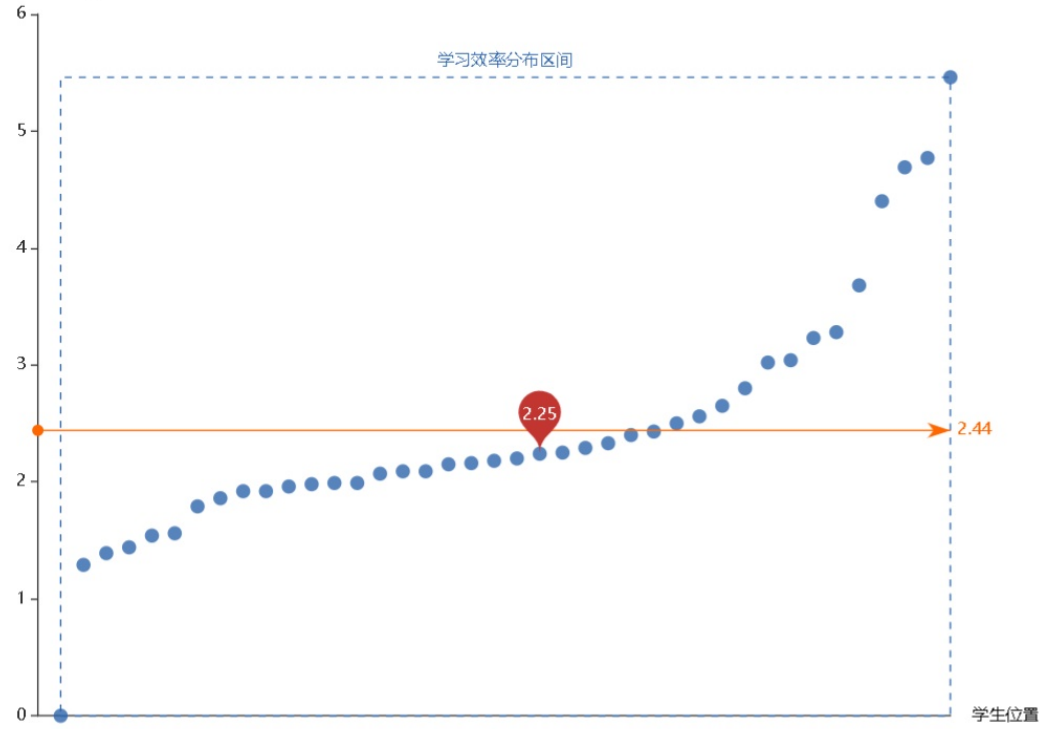
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 10:18	98	5	2024-04-23 15:08	33分 40秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-04-23 10:34	98	3	2024-04-23 15:08	44分 22秒	否	100/100	16.66/16.66	16.66
3	销毁线性表	2024-04-23 11:19	98	1	2024-04-23 15:08	4分 21秒	否	100/100	16.66/16.66	16.66
4	查找	2024-04-23 11:19	98	1	2024-04-23 15:09	21秒	否	100/100	10.66/16.66	10.66
5	删除	2024-04-23 11:19	98	1	2024-04-23 15:09	24秒	否	100/100	16.66/16.66	16.66
6	顺序表应用	2024-04-23 11:19	98	3	2024-04-23 15:09	10分 47秒	否	300/300	8.70/16.7	8.7

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	厉耀阳
学号	2340850052
学习效率	2.25
当前排名	18

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

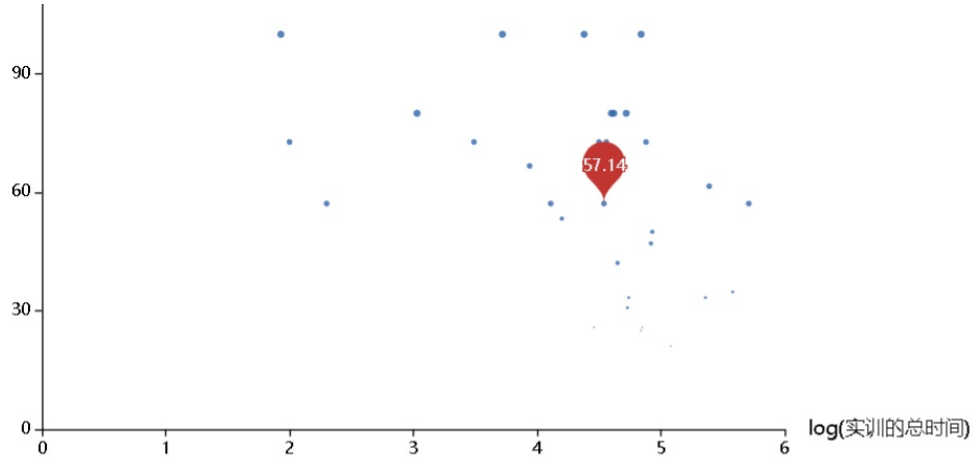
能力值

能力值（实训获得经验值/实训评测次数）



姓名	厉耀阳
学号	2340850052
能力	57.14

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建空顺序线性表

调分：16.66

学员代码 | 代码修改行数 98 | src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	559.68	2024-04-23 11:28	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	3166.65	2024-04-23 11:46	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0	466.62	2024-04-23 11:48	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0	3527.4	2024-04-23 11:53	查看代码
5	评测通过	0.01	88.2	2024-04-23 15:08	查看代码

代码评价

一键AI生成 编辑

75

暂无评价~

第2关 顺序线性表插入

调分：16.66

学员代码 代码修改行数 98 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
  int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
  int curNum;//用于存放顺序线性表中数据元素的个数 整型 curNum
  DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	2624.55	2024-04-23 11:18	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	638.85	2024-04-23 11:28	查看代码
3	评测通过	0.03	454.59	2024-04-23 15:08	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第3关 销毁线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 98　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	454.59	2024-04-23 15:08	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找

调分：10.66

学员代码 | 代码修改行数 98 | src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	318.5	2024-04-23 15:09	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 删除

调分：16.66

学员代码 ⓘ　　代码修改行数 98　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1220	2024-04-23 15:09	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 顺序表应用

调分： 8.7

学员代码 代码修改行数 98 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	342.52	2024-04-23 11:20	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	335.6	2024-04-23 11:36	查看代码
3	评测通过	0.01	148.14	2024-04-23 15:09	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



刘宣宣 按时通关

学号：2340850065　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-30 11:53	页面停留时长	1时 0分 41秒	14	- -	- -	89.0/100.0	良好

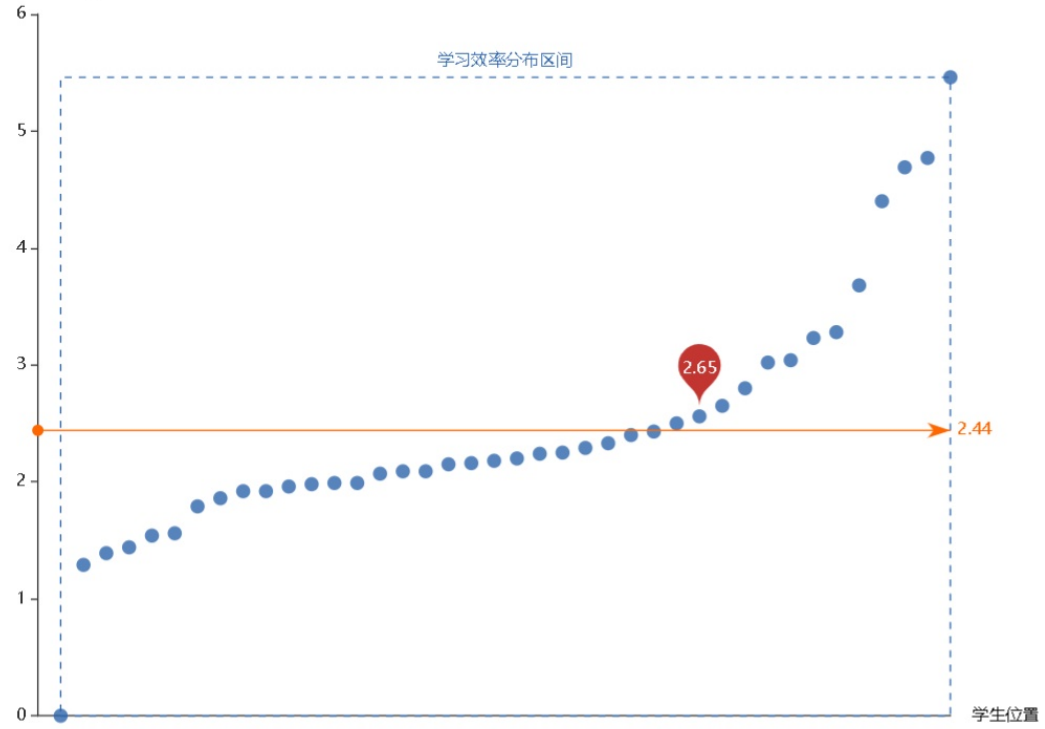
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 11:50	119	2	2024-04-24 15:39	8分 20秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-04-23 11:51	116	3	2024-04-30 11:52	32分 1秒	否	100/100	16.66/16.66	16.66
3	销毁线性表	2024-04-23 11:51	119	2	2024-04-24 15:49	2分 51秒	否	100/100	16.66/16.66	16.66
4	查找	2024-04-23 11:51	119	2	2024-04-24 15:50	4分 37秒	否	100/100	11.66/16.66	11.66
5	删除	2024-04-23 11:51	119	2	2024-04-24 15:49	30秒	否	100/100	14.66/16.66	14.66
6	顺序表应用	2024-04-23 11:51	118	3	2024-04-30 11:53	12分 22秒	否	300/300	12.70/16.7	12.7

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	刘宣宣
学号	2340850065
学习效率	2.65
当前排名	11

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

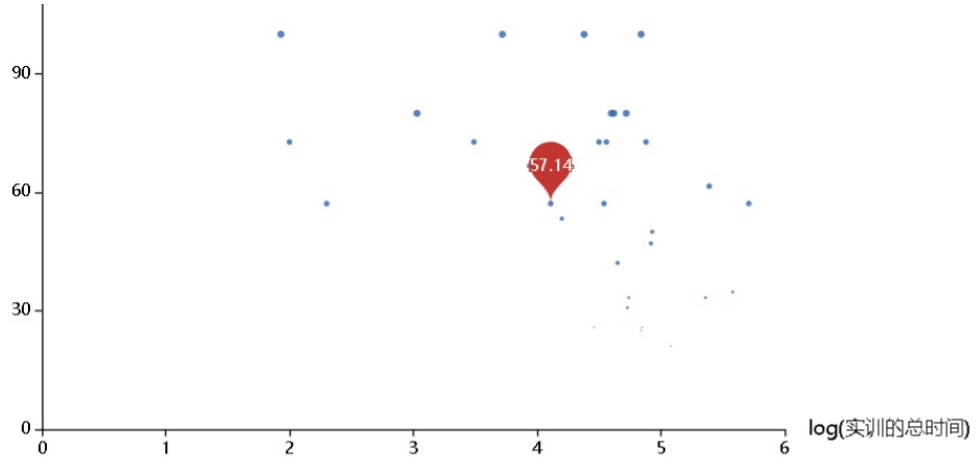
能力值

能力值（实训获得经验值/实训评测次数）



姓名	刘宣宣
学号	2340850065
能力	57.14

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建空顺序线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	487.95	2024-04-24 15:26	查看代码
2	评测通过	0.03	515.76	2024-04-24 15:39	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 顺序线性表插入

调分：16.66

学员代码 ⓘ　　代码修改行数 116　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	2918.67	2024-04-24 15:26	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.02	515.76	2024-04-24 15:49	查看代码
3	评测通过	0.02	1597.14	2024-04-30 11:52	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第3关 销毁线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.03	541.17	2024-04-24 15:30	查看代码
2	评测通过	0.05	3344.73	2024-04-24 15:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找

调分：11.66

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	368.6	2024-04-24 15:32	查看代码
2	评测通过	0.02	115.9	2024-04-24 15:50	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 删除

调分：14.66

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	177.91	2024-04-24 15:31	查看代码
2	评测通过	0.01	186.12	2024-04-24 15:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 顺序表应用

调分：12.7

学员代码 ⓘ　　代码修改行数 118　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType; //定义数据类型
struct seqList { //定义线性表结构体
    int MAXNUM; //用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum; //用于存放顺序线性表中数据元素的个数  整型 curNum
    DataType * element; //用于存放顺序线性表数据元素的连续空间的起始地址 指针类型
};

typedef struct seqList * PseqList; //定义线性表指针类型，方便后序的函数传递参数

//第一关：创建空的顺序线性表
PseqList createNullList_seq(int m) {
    //输入参数为能存放的最大元素个数
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2295.98	2024-04-24 15:31	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	309	2024-04-24 15:50	查看代码
3	评测通过	0.03	2439.74	2024-04-30 11:53	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



张婷婷 按时通关

学号：2340850058　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-30 10:59	页面停留时长	51分 36秒	12	- -	- -	78.0/100.0	良好

阶段成绩

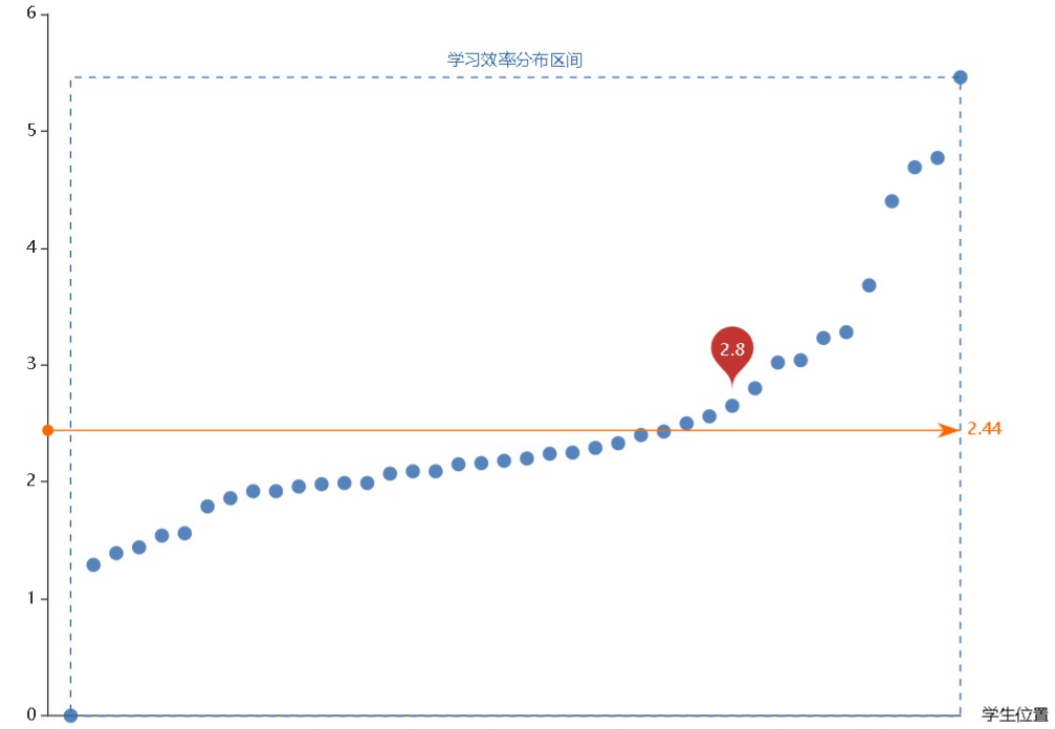
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 11:35	78	1	2024-04-23 11:51	6分 37秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-04-23 11:36	95	3	2024-04-30 10:59	11分 28秒	否	100/100	8.66/16.66	8.66
3	销毁线性表	2024-04-23 11:42	78	1	2024-04-23 11:52	1分 18秒	否	100/100	16.66/16.66	16.66
4	查找	2024-04-23 11:42	78	1	2024-04-23 11:53	1分 26秒	否	100/100	16.66/16.66	16.66
5	删除	2024-04-23 11:42	78	1	2024-04-23 11:54	49秒	否	100/100	10.66/16.66	10.66
6	顺序表应用	2024-04-23 11:42	92	5	2024-04-30 10:51	29分 58秒	否	300/300	8.70/16.7	8.7

图形统计

90

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	张婷婷
学号	2340850058
学习效率	2.8
当前排名	10

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

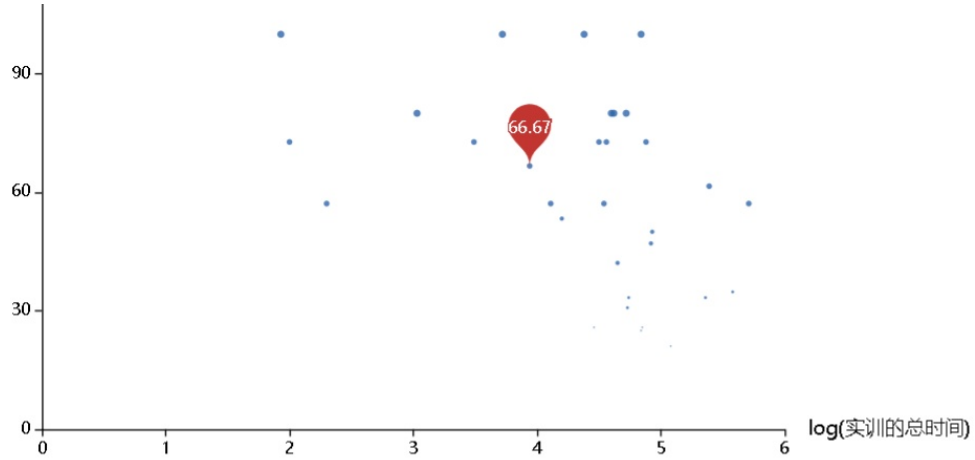
能力值

能力值（实训获得经验值/实训评测次数）



姓名	张婷婷
学号	2340850058
能力	66.67

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建空顺序线性表

调分: 16.66

学员代码 代码修改行数 78 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	574.08	2024-04-23 11:51	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 顺序线性表插入

调分： 8.66

学员代码 代码修改行数 95 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.02	506.31	2024-04-23 11:54	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.01	590.73	2024-04-23 11:58	查看代码
3	评测通过	0.03	477.06	2024-04-30 10:59	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 销毁线性表

调分: 16.66

学员代码 代码修改行数 78 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	1887.63	2024-04-23 11:52	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 查找

调分: 16.66

学员代码 ⓘ　　代码修改行数 78　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	1258.42	2024-04-23 11:53	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 删除

调分: 10.66

学员代码 代码修改行数 78 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	168.77	2024-04-23 11:54	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 顺序表应用

调分： 8.7

学员代码 代码修改行数 92 src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
  int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
  int curNum;//用于存放顺序线性表中数据元素的个数 整型 curNum
  DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	96.2	2024-04-23 11:54	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	885.18	2024-04-30 10:43	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	947.12	2024-04-30 10:46	查看代码
4	评测通过	0.02	1606.46	2024-04-30 10:51	查看代码
5	评测通过	0.02	316.02	2024-04-30 10:54	查看代码

代码评价 一键AI生成 编辑

98

暂无评价~

总体评价

点评作业



吴杰 补交通关

学号：2340850075　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-12 00:03	页面停留时长	2时 28分 15秒	6	- -	18	70.0/100.0	良好

阶段成绩

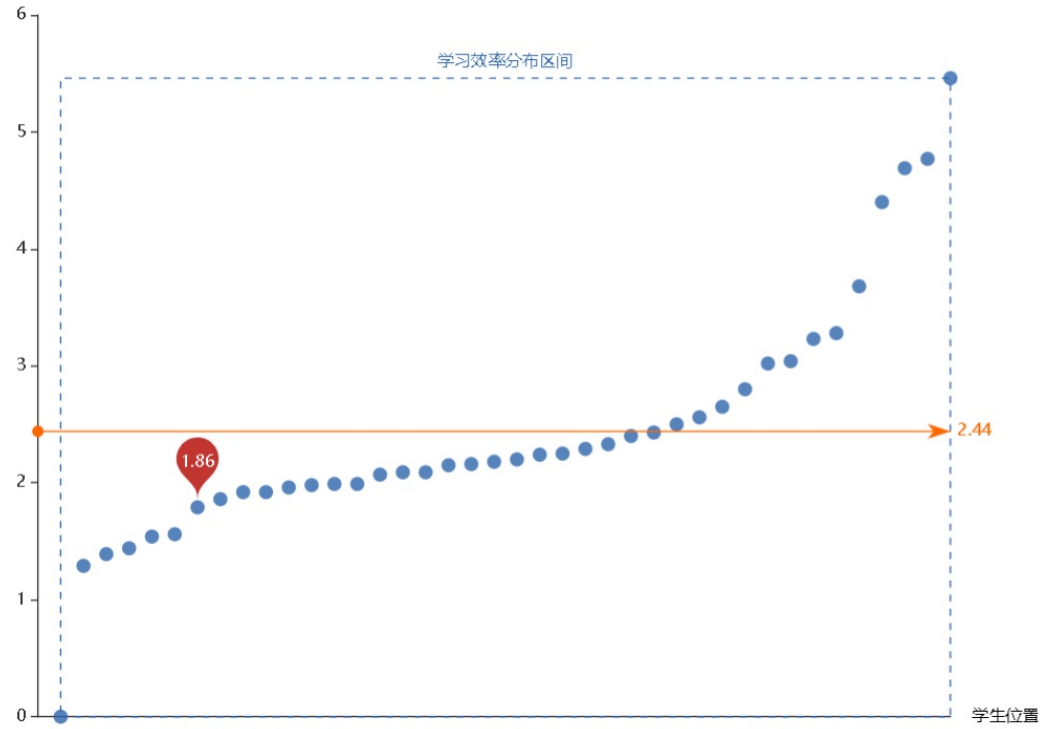
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空顺序线性表	2024-04-23 10:33	88	1	2024-06-12 00:02	2时 27分 44秒	否	100/100	16.66/16.66	16.66
2	顺序线性表插入	2024-06-12 00:02	88	1	2024-06-12 00:02	4秒	否	100/100	16.66/16.66	16.66
3	销毁线性表	2024-06-12 00:02	88	1	2024-06-12 00:02	9秒	否	100/100	16.66/16.66	16.66
4	查找	2024-06-12 00:03	88	1	2024-06-12 00:03	4秒	否	100/100	10.66/16.66	10.66
5	删除	2024-06-12 00:03	88	1	2024-06-12 00:03	8秒	否	100/100	10.66/16.66	10.66
6	顺序表应用	2024-06-12	88	1	99 2024-06-12	6秒	否	300/30	16.70/16.7	16.7

	00:03	00:03			0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	吴杰
学号	2340850075
学习效率	1.86
当前排名	33

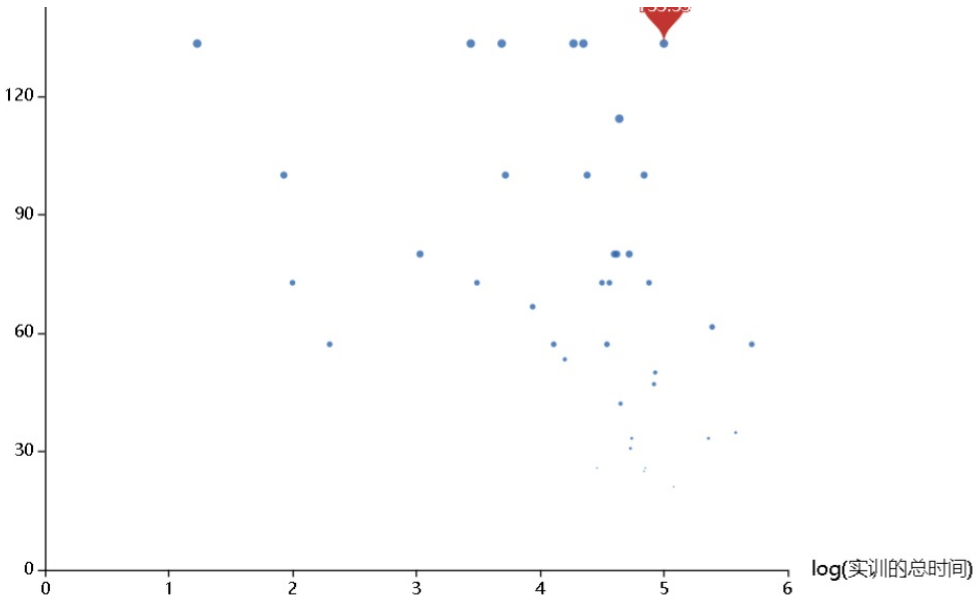
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）



姓名	吴杰
学号	2340850075
能力	133.33



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空顺序线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 88　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0	3659.55	2024-06-12 00:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 顺序线性表插入

调分：16.66

学员代码 ⓘ　　代码修改行数 88　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	166.47	2024-06-12 00:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 销毁线性表

调分：16.66

学员代码 ⓘ　　代码修改行数 88　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.02	473.67	2024-06-12 00:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找

调分：10.66

学员代码 | 代码修改行数 88 | src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.03	2439.7	2024-06-12 00:03	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 删除

调分：10.66

学员代码 ⓘ　　代码修改行数 88　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.02	1219.83	2024-06-12 00:03	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 顺序表应用

调分：16.7

学员代码 ⓘ　　代码修改行数 88　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

/*此处是顺序线性表数据结构定义*/
typedef int DataType;
struct seqList
{//有3个数据成员
    int MAXNUM;//用于记录顺序线性表中能存放的最大元素个数的 整型 MAXNUM
    int curNum;//用于存放顺序线性表中数据元素的个数  整型  curNum
    DataType *element;//用于存放顺序线性表数据元素的连续空间的起始地址
};

typedef struct seqList *PseqList;
//第一关
PseqList createNullList_seq(int m)
{//此处填写代码，创建一个空的顺序线性表，能存放的最大元素个数为 m
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.03	110.98	2024-06-12 00:03	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：5/5　　最新完成关卡：5/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-30 10:53	页面停留时长	56分 10秒	8	- -	- -	94.0/100.0	优秀

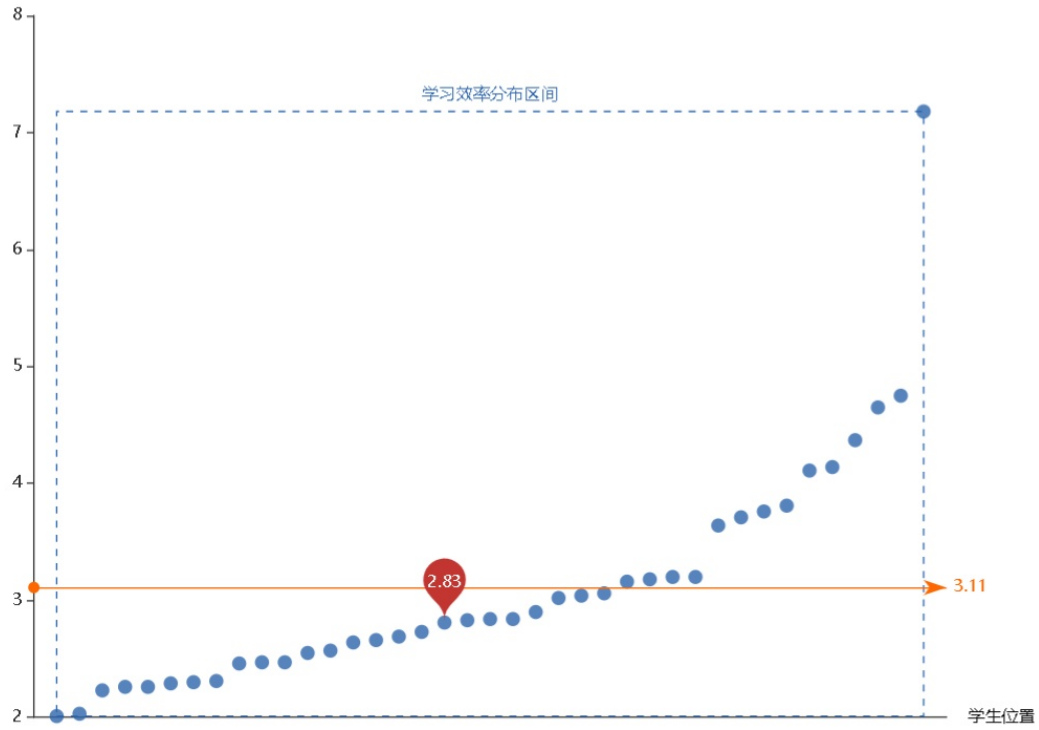
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-29 18:22	16	2	2024-04-29 18:33	11分 55秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-29 18:34	36	3	2024-04-29 18:47	13分 27秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-29 18:45	50	1	2024-04-30 10:33	11分 48秒	否	300/300	18.00/20	18
4	查找和交换	2024-04-30 10:34	88	1	2024-04-30 10:49	14分 50秒	否	300/300	16.00/20	16
5	链表销毁	2024-04-30 10:48	99	1	2024-04-30 10:53	4分 10秒	否	100/100	20.00/20	20

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

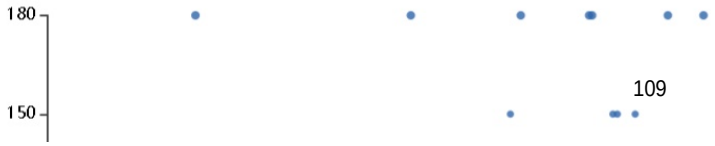


姓名	李雯君
学号	2340850043
学习效率	2.83
当前排名	21

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

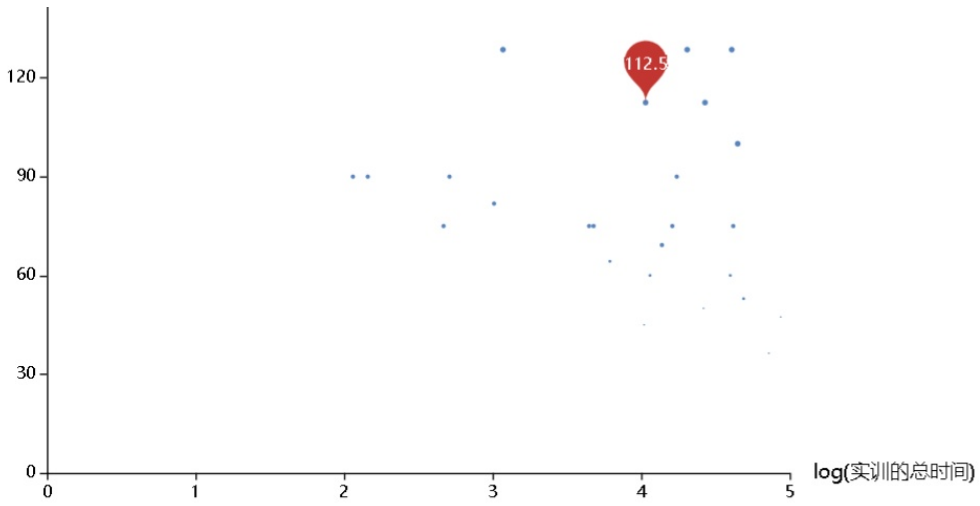
能力值

能力值（实训获得经验值/实训评测次数）



姓名	李雯君
学号	2340850043
能力	112.5

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值



实训详情

第1关 链表数据结构定义及初始化链表

调分: 20

学员代码 代码修改行数 16 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码


struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	1219.64	2024-04-29 18:28	查看代码
2	评测通过	0.02	76.72	2024-04-29 18:33	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分: 20

学员代码 代码修改行数 36 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码


struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateLis()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	59.41	2024-04-29 18:45	查看代码
2	评测通过	0.01	65.26	2024-04-29 18:47	查看代码
3	评测通过	0	66.5	2024-04-29 18:51	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分：18

学员代码 | 代码修改行数 50 | src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码


struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	461.08	2024-04-30 10:33	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 查找和交换

调分: 16

学员代码 代码修改行数 88 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码


struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1219.73	2024-04-30 10:49	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 链表销毁

调分: 20

学员代码 | 代码修改行数 99 | src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码


struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	2439.58	2024-04-30 10:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：5/5　　最新完成关卡：5/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-04 20:24	页面停留时长	1时 39分 10秒	15	- -	- -	95.0/100.0	优秀

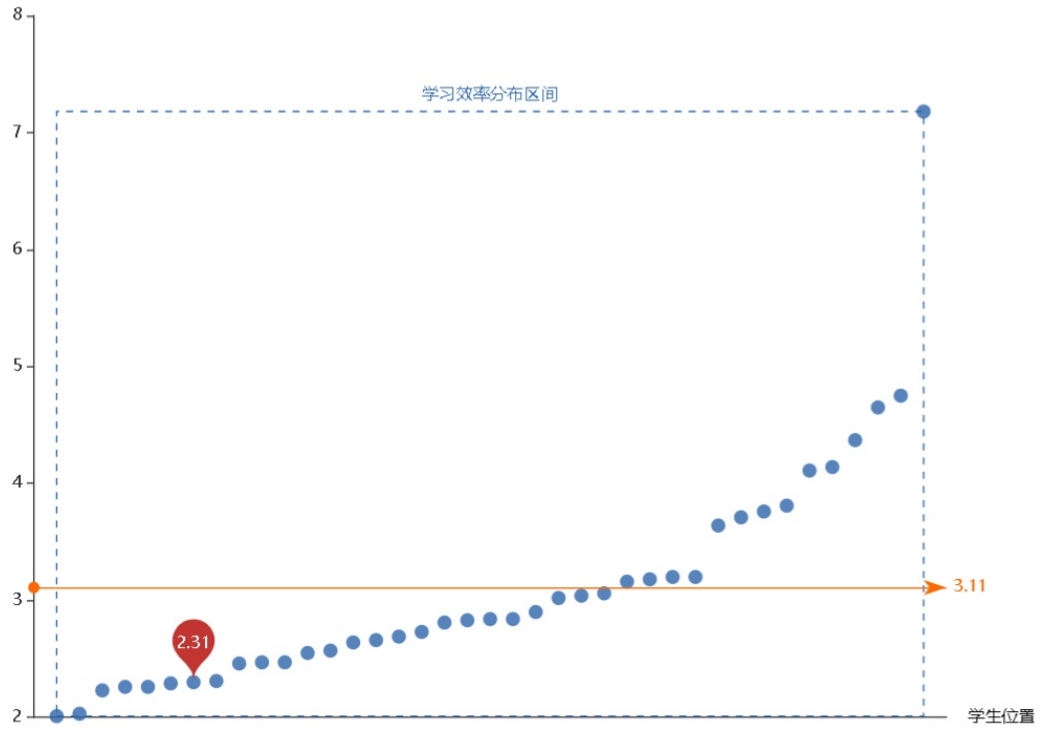
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-30 10:24	7	9	2024-04-30 10:44	19分 50秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-30 10:49	25	2	2024-04-30 11:42	52分 2秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-30 11:43	36	1	2024-04-30 11:51	7分 59秒	否	300/300	16.00/20	16
4	查找和交换	2024-05-04 20:00	56	1	2024-05-04 20:20	15分 44秒	否	300/300	19.00/20	19
5	链表销毁	2024-05-04 20:21	66	2	2024-05-04 20:24	3分 35秒	否	100/100	20.00/20	20

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

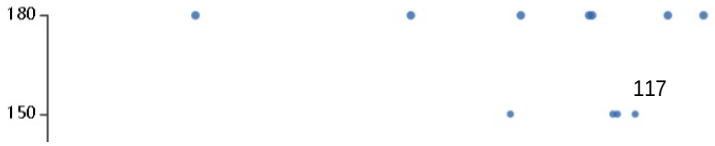


姓名	黄心媛
学号	2340850073
学习效率	2.31
当前排名	32

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

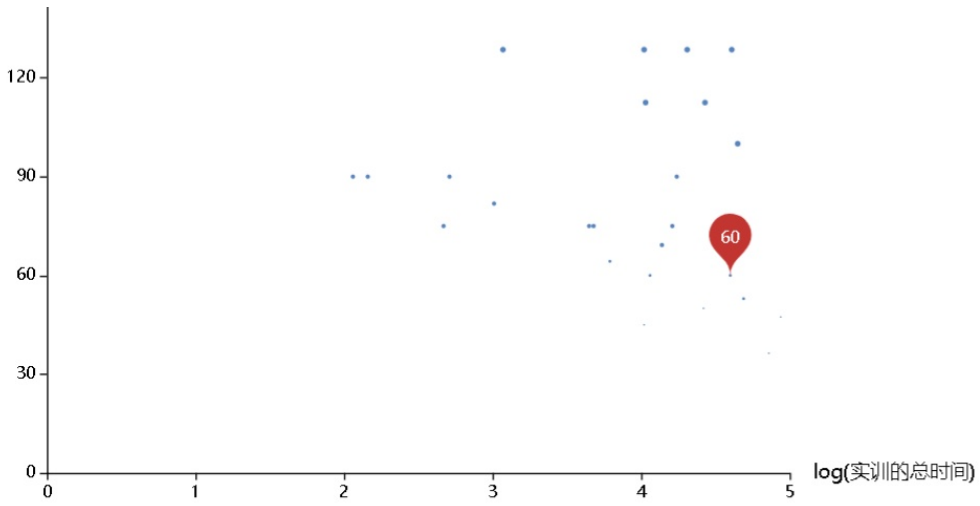
能力值

能力值（实训获得经验值/实训评测次数）



姓名	黄心媛
学号	2340850073
能力	60

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值



实训详情

第1关 链表数据结构定义及初始化链表

调分：20

学员代码 ⓘ　　代码修改行数 7　　src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node *head=(struct node *)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
  return head;
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	152.77	2024-04-30 10:31	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	160.01	2024-04-30 10:33	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0	408.73	2024-04-30 10:41	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0	45.75	2024-04-30 10:42	查看代码
5	评测通过	0.01	845.74	2024-04-30 10:44	查看代码
6	评测通过	0.01	846.6	2024-04-30 10:47	查看代码
7	共有1组测试集，其中有1组测试结果不匹配	0	795.54	2024-04-30 10:48	查看代码

8	评测通过	0.01	155.82	2024-04-30 10:48	查看代码
9	评测通过	0.01	846.6	2024-04-30 10:50	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分: 20

学员代码 代码修改行数 25 src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head=(struct node *)malloc(sizeof(struct node));
    head->data=0;
    head->next=NULL;
    return head;
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1219.87	2024-04-30 11:42	查看代码
2	评测通过	0.02	1219.79	2024-04-30 11:42	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分: 16

学员代码 代码修改行数 36 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head=(struct node *)malloc(sizeof(struct node));
    head->data=0;
    head->next=NULL;
    return head;
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	141.96	2024-04-30 11:51	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 查找和交换

调分: 19

学员代码 代码修改行数 56 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head=(struct node *)malloc(sizeof(struct node));
    head->data=0;
    head->next=NULL;
    return head;
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	158.55	2024-05-04 20:20	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 链表销毁

调分: 20

学员代码 代码修改行数 66 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head=(struct node *)malloc(sizeof(struct node));
    head->data=0;
    head->next=NULL;
    return head;
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	317.1	2024-05-04 20:23	查看代码
2	评测通过	0.03	98.22	2024-05-04 20:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



张佳慧 按时通关

学号：2340850044　　分班：未分班

截止前完成关卡：5/5　　最新完成关卡：5/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-04-29 18:37	页面停留时长	22分 40秒	6	- -	- -	89.0/100.0	良好

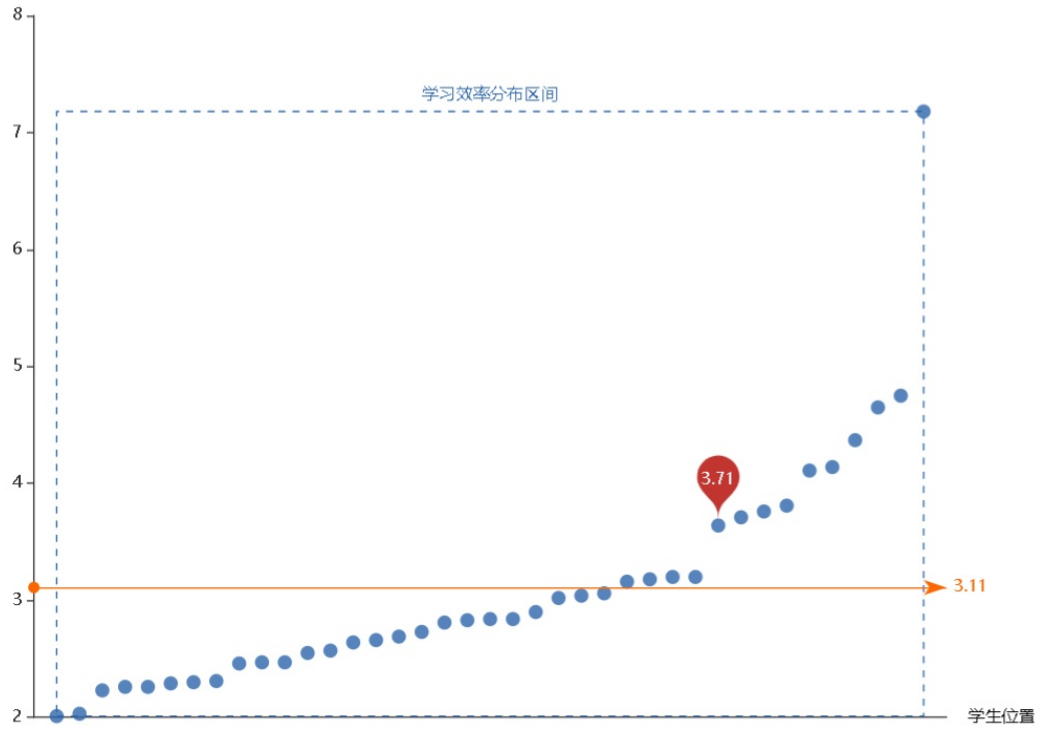
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-29 18:14	7	1	2024-04-29 18:22	7分 41秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-29 18:22	23	1	2024-04-29 18:29	6分 37秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-29 18:29	33	1	2024-04-29 18:31	2分 22秒	否	300/300	20.00/20	20
4	查找和交换	2024-04-29 18:31	51	1	2024-04-29 18:35	4分 4秒	否	300/300	15.00/20	15
5	链表销毁	2024-04-29 18:35	62	2	2024-04-29 18:37	1分 56秒	否	100/100	14.00/20	14

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

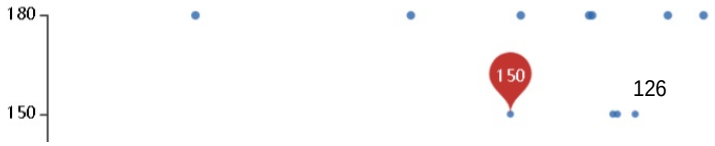


姓名	张佳慧
学号	2340850044
学习效率	3.71
当前排名	9

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

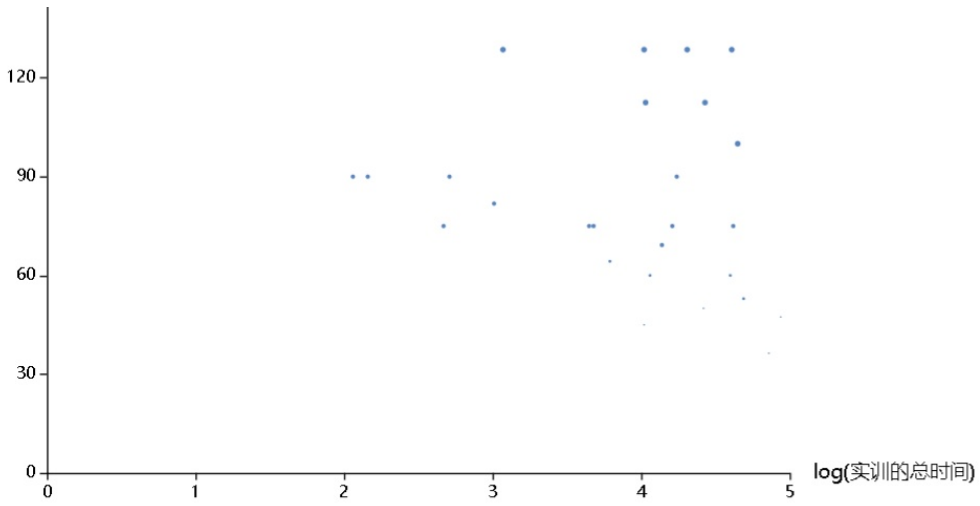
能力值（实训获得经验值/实训评测次数）



姓名	张佳慧
学号	2340850044
能力	150

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值

小圆圈越大，评测次数越少



实训详情

第1关 链表数据结构定义及初始化链表

调分: 20

学员代码 代码修改行数 7 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
typedef struct node*ph;
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  ph head = (ph)malloc(sizeof(struct node));
  head->data = 0;
  head->next = NULL;
  return head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	47.21	2024-04-29 18:22	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分: 20

学员代码 代码修改行数 23 src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
typedef struct node*ph;
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  ph head = (ph)malloc(sizeof(struct node));
  head->data = 0;
  head->next = NULL;
  return head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	58.95	2024-04-29 18:29	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分：20

学员代码 ⓘ　　代码修改行数 33　　src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
typedef struct node*ph;
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  ph head = (ph)malloc(sizeof(struct node));
  head->data = 0;
  head->next = NULL;
  return head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	119	2024-04-29 18:31	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找和交换

调分：15

学员代码 ⓘ　　代码修改行数 51　　src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
typedef struct node*ph;
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  ph head = (ph)malloc(sizeof(struct node));
  head->data = 0;
  head->next = NULL;
  return head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1173.03	2024-04-29 18:35	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第5关 链表销毁

调分：14

学员代码 代码修改行数 62 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node *next;
};
typedef struct node*ph;
struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  ph head = (ph)malloc(sizeof(struct node));
  head->data = 0;
  head->next = NULL;
  return head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	119	2024-04-29 18:37	查看代码
2	评测通过	0.02	87.4	2024-04-29 18:37	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



罗贤文 按时通关

学号：2340850072　　分班：未分班

截止前完成关卡：5/5　　最新完成关卡：5/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-07 10:23	页面停留时长	38分 20秒	12	- -	- -	89.0/100.0	良好

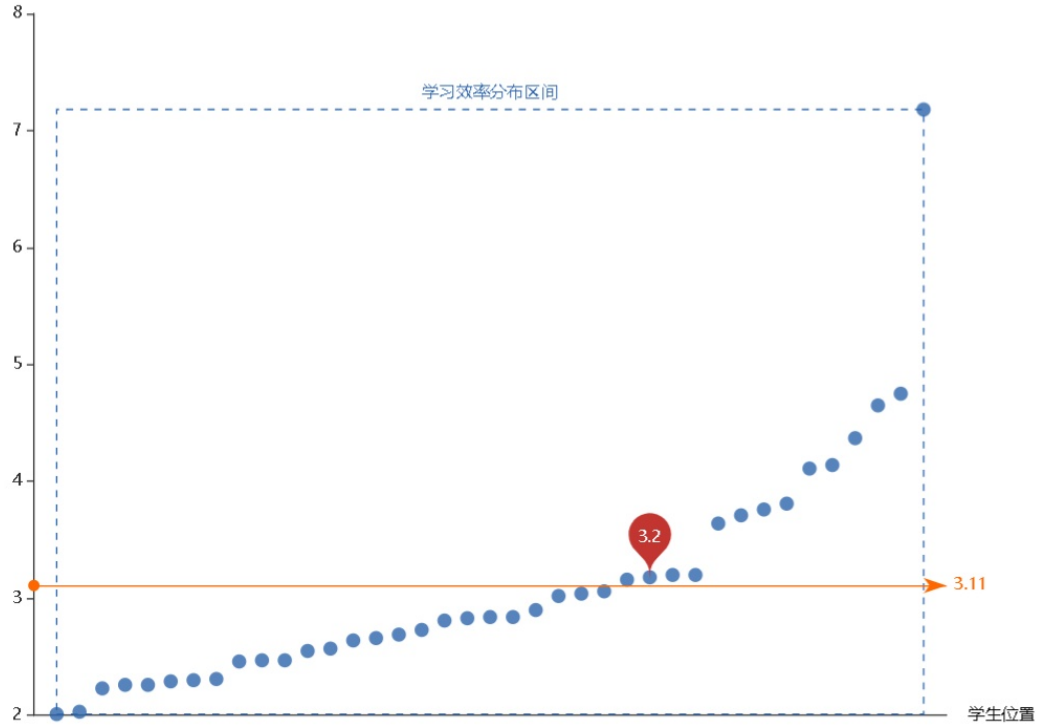
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-28 18:54	7	1	2024-04-28 18:58	4分 7秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-28 18:58	26	3	2024-04-28 19:05	6分 43秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-28 19:05	42	2	2024-05-07 10:01	6分 6秒	否	300/300	20.00/20	20
4	查找和交换	2024-05-07 10:01	56	5	2024-05-07 10:12	10分 22秒	否	300/300	14.00/20	14
5	链表销毁	2024-05-07 10:12	67	1	2024-05-07 10:23	11分 2秒	否	100/100	15.00/20	15

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

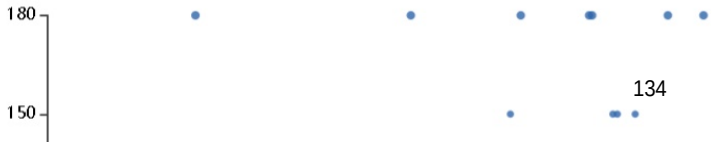


姓名	罗贤文
学号	2340850072
学习效率	3.2
当前排名	12

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

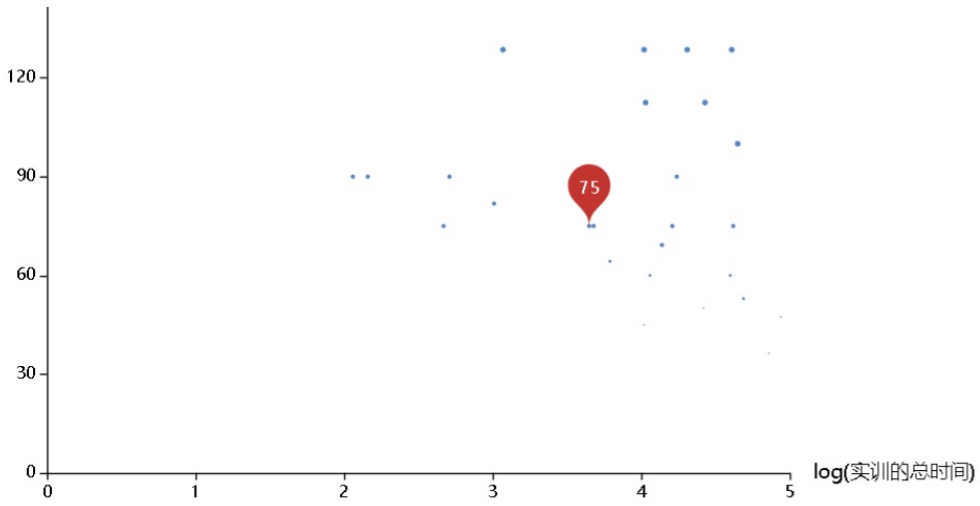
能力值（实训获得经验值/实训评测次数）



姓名	罗贤文
学号	2340850072
能力	75

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值

小圆圈越大，评测次数越少



实训详情

第1关 链表数据结构定义及初始化链表

调分: 20

学员代码 | 代码修改行数 7 | src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    struct node *next;
    int data;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node* head;
  head=(struct node*)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	211.27	2024-04-28 18:58	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分: 20

学员代码 | 代码修改行数 26 | src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    struct node *next;
    int data;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node* head;
  head=(struct node*)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	169.05	2024-04-28 19:04	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	158.66	2024-04-28 19:05	查看代码
3	评测通过	0	211.27	2024-04-28 19:05	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分: 20

学员代码 代码修改行数 42 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    struct node *next;
    int data;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node* head;
  head=(struct node*)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.03	292.56	2024-04-28 19:08	查看代码
2	评测通过	0.01	85.58	2024-05-07 10:01	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 查找和交换

调分：14

学员代码 ⓘ　　代码修改行数 56　　src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    struct node *next;
    int data;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node* head;
  head=(struct node*)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.03	150.95	2024-05-07 10:09	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	150.95	2024-05-07 10:10	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0.01	59.96	2024-05-07 10:10	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0.01	164.8	2024-05-07 10:11	查看代码
5	评测通过	0.01	59.96	2024-05-07 10:12	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 链表销毁

调分: 15

学员代码 | 代码修改行数 67 | src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    struct node *next;
    int data;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
  struct node* head;
  head=(struct node*)malloc(sizeof(struct node));
  head->data=0;
  head->next=NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	2440.34	2024-05-07 10:23	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



高林松 未通关

学号：2340850063　　分班：未分班

截止前完成关卡：4/5　　最新完成关卡：4/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
--	页面停留时长	2时 1分 26秒	22	- -	- -	70.0/100.0	良好

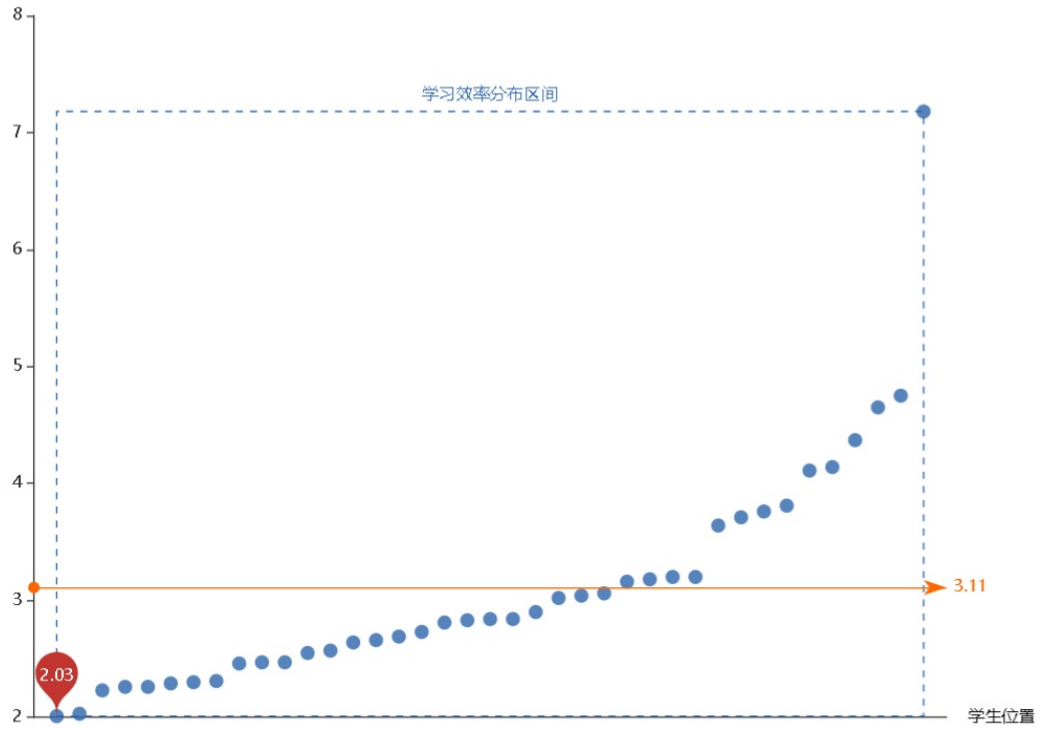
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-23 11:36	6	2	2024-04-30 10:40	33分 37秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-30 10:43	25	7	2024-04-30 11:01	17分 30秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-30 11:01	44	2	2024-05-07 10:42	1时 9分 46秒	否	300/300	20.00/20	20
4	查找和交换	2024-05-07 10:42	74	3	2024-05-07 10:42	33秒	否	300/300	10.00/20	10
5	链表销毁	2024-05-07 10:42	- -	8	--	--	否	0/100	0.00/20	0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

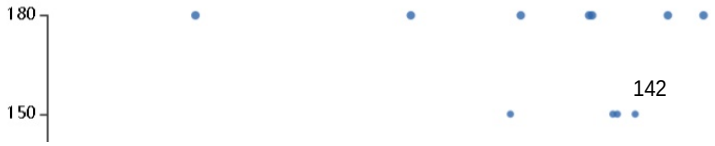


姓名	高林松
学号	2340850063
学习效率	2.03
当前排名	38

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

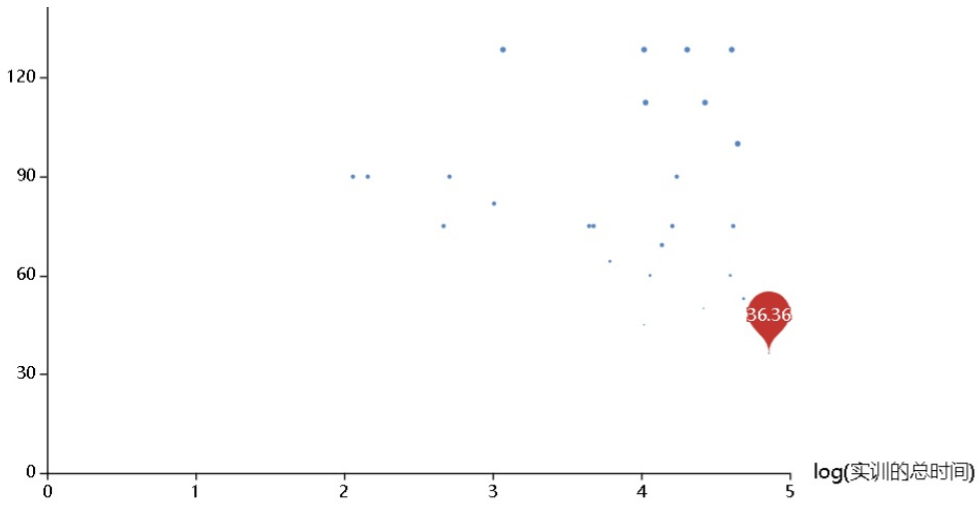
能力值

能力值（实训获得经验值/实训评测次数）



姓名	高林松
学号	2340850063
能力	36.36

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值



实训详情

第1关 链表数据结构定义及初始化链表

调分：20

学员代码 ⓘ　　代码修改行数 6　　src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data ;
    struct node *next ;

};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head = (struct node *)malloc(sizeof(struct node));
    head->data = 0;
    head->next = NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	473.56	2024-04-30 10:31	查看代码
2	评测通过	0.01	49.61	2024-04-30 10:40	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分：20

学员代码 代码修改行数 25 src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data ;
    struct node *next ;

};

struct node *mycreateLis()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head = (struct node *)malloc(sizeof(struct node));
    head->data = 0;
    head->next = NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	451.66	2024-04-30 10:43	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	803.23	2024-04-30 10:45	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0.01	473.56	2024-04-30 10:50	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0.02	803.23	2024-04-30 10:51	查看代码
5	共有1组测试集，其中有1组测试结果不匹配	0	451.66	2024-04-30 10:55	查看代码
6	共有1组测试集，其中有1组测试结果不匹配	0.01	435.8	2024-04-30 10:56	查看代码
7	评测通过	0.01	454.07	2024-04-30 11:01	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分：20

学员代码 ⓘ　　代码修改行数 44　　src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data ;
    struct node *next ;

};

struct node *mycreateLis()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *head = (struct node *)malloc(sizeof(struct node));
    head->data = 0;
    head->next = NULL;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	91.36	2024-04-30 11:06	查看代码
2	评测通过	0.01	360.62	2024-05-07 10:42	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 查找和交换

调分： 10

学员代码 代码修改行数 74 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node* next;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node* list = (struct node*)malloc(sizeof(struct node));
    list->next = NULL;
    list->data = 0;

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	174.22	2024-05-07 10:42	查看代码
2	评测通过 截止后评测	0.01	1072.19	2024-06-11 10:26	查看代码
3	评测通过 截止后评测	0	298.96	2024-06-11 10:31	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 链表销毁

调分： 0

学员代码 src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
    struct node* next;
};

struct node *mycreateList()
{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node* list = (struct node*)malloc(sizeof(struct node));
    list->next = NULL;
    list->data = 0;

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.05	336.04	2024-05-07 10:43	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.04	360.62	2024-05-07 10:44	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	2032.64	2024-05-07 10:45	查看代码
4	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0.04	597.92	2024-06-11 10:26	查看代码
5	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0.03	597.92	2024-06-11 10:27	查看代码
6	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	2144.38	2024-06-11 10:28	查看代码
7	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0.04	2144.38	2024-06-11 10:31	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	截止后评测	0.07	313.3	2024-06-11 10:31	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



范淇越 补交通关

学号：2340850078　　分班：未分班

截止前完成关卡：5/5　　最新完成关卡：5/5　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-11 11:53	页面停留时长	20分 18秒	11	- -	9	74.0/100.0	良好

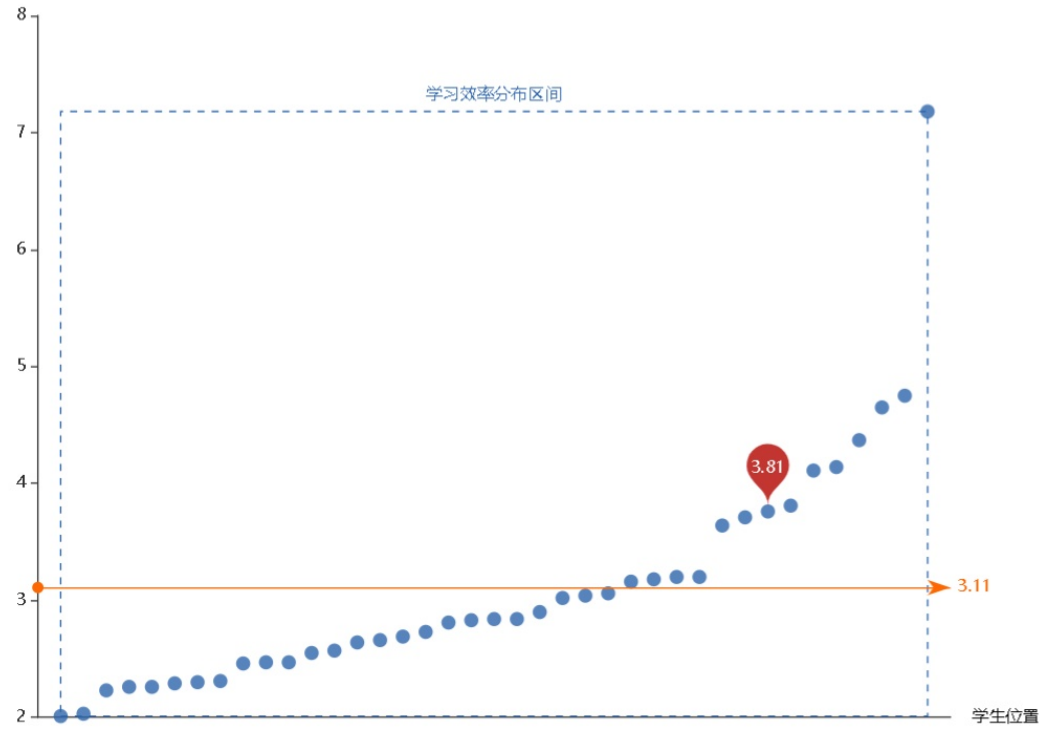
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	链表数据结构定义及初始化链表	2024-04-23 16:40	10	1	2024-04-23 16:46	6分 22秒	否	100/100	20.00/20	20
2	链表插入和遍历链表输出链表的操作	2024-04-23 16:46	43	1	2024-04-23 16:48	2分 12秒	否	100/100	20.00/20	20
3	链表逆置	2024-04-23 16:49	131	7	2024-06-11 11:53	11分 35秒	否	300/300	17.00/20	17
4	查找和交换	2024-06-11 11:53	131	1	2024-06-11 11:53	5秒	否	300/300	10.00/20	10
5	链表销毁	2024-06-11 11:53	131	1	2024-06-11 11:53	4秒	否	100/100	16.00/20	16

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)

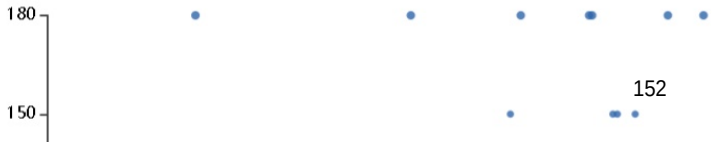


姓名	范淇越
学号	2340850078
学习效率	3.81
当前排名	7

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

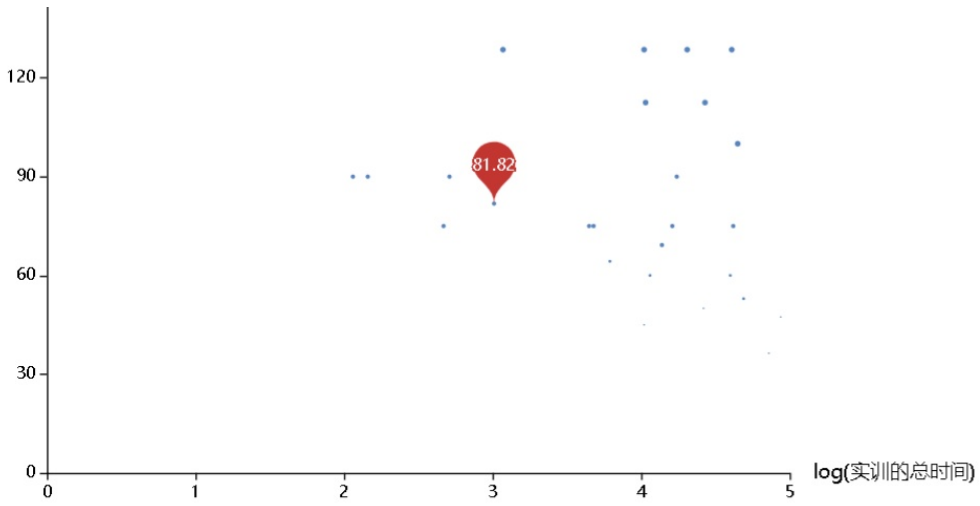
能力值

能力值（实训获得经验值/实训评测次数）



姓名	范淇越
学号	2340850078
能力	81.82

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值



实训详情

第1关 链表数据结构定义及初始化链表

调分: 20

学员代码 ⓘ　　代码修改行数 10　　src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
     struct node* next;
};
typedef struct node *PLIST;
typedef struct node NODE;
struct node *mycreateLis()


{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
   struct node *li = (struct node*)malloc(sizeof(struct node));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	467.49	2024-04-23 16:46	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 链表插入和遍历链表输出链表的操作

调分: 20

学员代码 代码修改行数 43 src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
//第一关代码

struct node
{//此处填写代码，定义链表结点类型，包含一个存放整型数据的 data 成员，和一个指向下一个结点的next成员
    int data;
     struct node* next;
};
typedef struct node *PLIST;
typedef struct node NODE;
struct node *mycreateLis()


{//此处填写代码，创建一个只有一个头结点的空链表，头节点的数据域赋值为0，并将表头结点的地址返回
    struct node *li = (struct node*)malloc(sizeof(struct node));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	90.46	2024-04-23 16:48	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 链表逆置

调分：17

学员代码 | 代码修改行数 131 | src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>

// 第一关代码

struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0.01	313.42	2024-06-11 11:42	查看代码
2	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0.02	105.06	2024-06-11 11:43	查看代码
3	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	314.68	2024-06-11 11:44	查看代码
4	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	313.42	2024-06-11 11:45	查看代码
5	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	1095.22	2024-06-11 11:47	查看代码
6	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	433.8	2024-06-11 11:51	查看代码
7	评测通过 截止后评测	0.02	433.8	2024-06-11 11:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 查找和交换

调分： 10

学员代码 代码修改行数 131 src/list.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>

// 第一关代码

struct node
{
    int data;            // 存放整型数据的 data 成员
    struct node *next;   // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	547.61	2024-06-11 11:53	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第5关 链表销毁

调分: 16

学员代码 | 代码修改行数 131 | src/list.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>

// 第一关代码

struct node
{
    int data;           // 存放整型数据的 data 成员
    struct node *next;  // 指向下一个结点的 next 成员
};

struct node *mycreateList()
{
    // 创建一个只有一个头结点的空链表
    // 头节点的数据域赋值为 0，并将表头结点的地址返回

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	433.8	2024-06-11 11:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-07 11:15	页面停留时长	1时 0分 31秒	13	- -	- -	96.0/100.0	优秀

阶段成绩

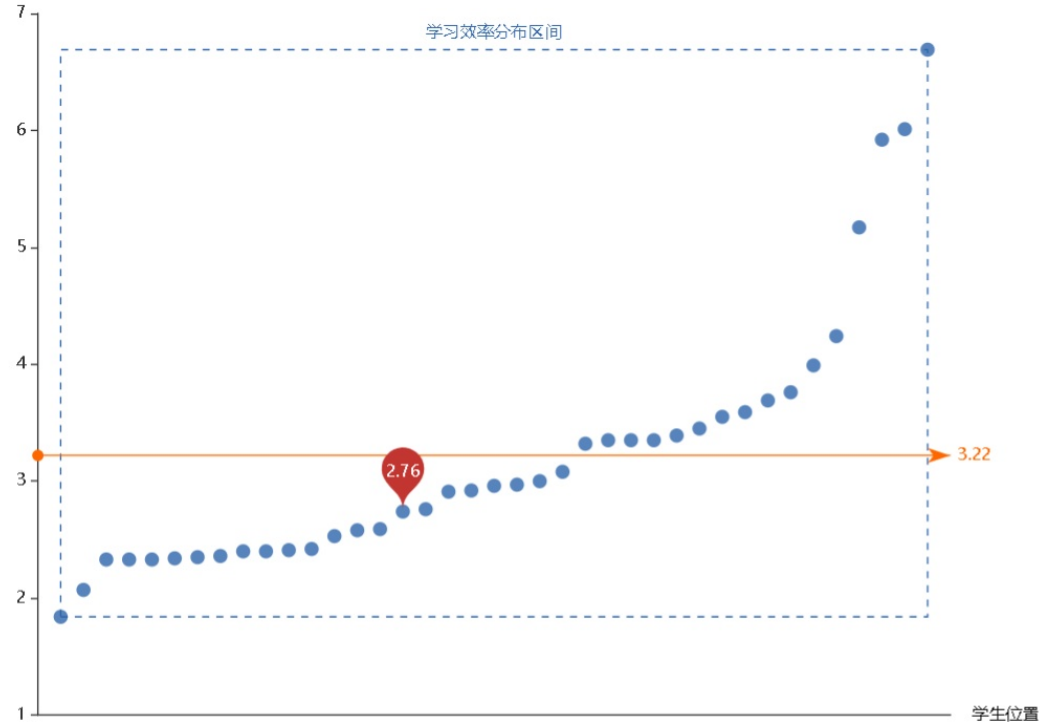
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-04-30 10:55	18	4	2024-05-07 10:26	13分 22秒	否	100/100	14.28/14.28	14.28
2	栈空判断	2024-05-07 10:24	33	1	2024-05-07 10:29	3分 18秒	否	100/100	14.28/14.28	14.28
3	栈满判断	2024-05-07 10:29	41	1	2024-05-07 10:31	2分 6秒	否	100/100	10.28/14.28	10.28
4	压栈	2024-05-07 10:31	52	2	2024-05-07 10:35	3分 36秒	否	100/100	14.28/14.28	14.28
5	弹栈	2024-05-07 10:34	65	2	2024-05-07 10:43	8分 19秒	否	100/100	14.28/14.28	14.28
6	取栈顶元素	2024-05-07	82	1	160 2024-05-07	4分 45秒	否	100/10	14.28/14.2	14.28

		10:44			10:49			0	8	
7	利用栈实现数进制转换	2024-05-07 10:49	144	2	2024-05-07 11:15	25分 5秒	否	300/300	14.32/14.32	14.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	李雯君
学号	2340850043
学习效率	2.76
当前排名	23

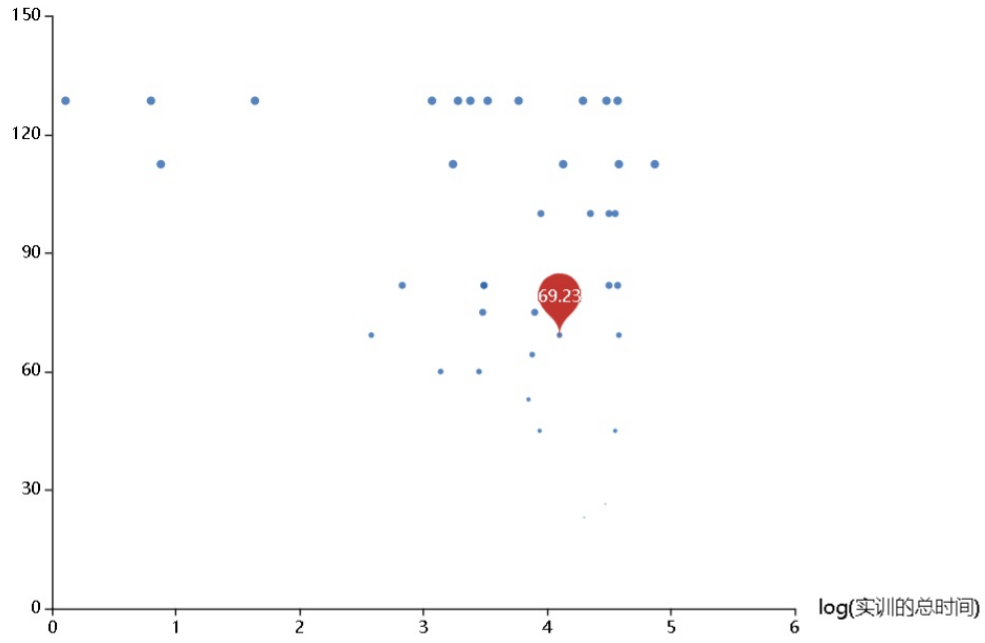
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043

能力 69.23



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：14.28

学员代码 ⓘ　　代码修改行数 18　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有1组测试结果不匹配	0.03	477.78	2024-05-07 10:24	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.04	477.78	2024-05-07 10:25	查看代码
3	共有3组测试集，其中有1组测试结果不匹配	0.05	474.33	2024-05-07 10:25	查看代码
4	评测通过	0.02	474.33	2024-05-07 10:26	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 栈空判断

调分: 14.28

学员代码 | 代码修改行数 33 | src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	2439.98	2024-05-07 10:29	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 栈满判断

调分: 10.28

学员代码 | 代码修改行数 41 | src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1219.99	2024-05-07 10:31	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 压栈

调分：14.28

学员代码 ⓘ　　代码修改行数 52　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有1组测试结果不匹配	0.02	394.6	2024-05-07 10:34	查看代码
2	评测通过	0.03	1809.76	2024-05-07 10:35	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 弹栈

调分：14.28

学员代码 ⓘ　　代码修改行数 65　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	2439.78	2024-05-07 10:43	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2439.78	2024-05-07 10:47	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 取栈顶元素

调分: 14.28

学员代码 ⓘ　　代码修改行数 82　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	1831.46	2024-05-07 10:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第7关 利用栈实现数进制转换

调分：14.32

学员代码 ⓘ　　代码修改行数 144　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{
    PseqStack p = (PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	0	2024-05-07 11:09	查看代码
2	评测通过	0.03	2606.49	2024-05-07 11:15	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-07 11:46	页面停留时长	1时 13分 24秒	39	- -	- -	97.0/100.0	优秀

阶段成绩

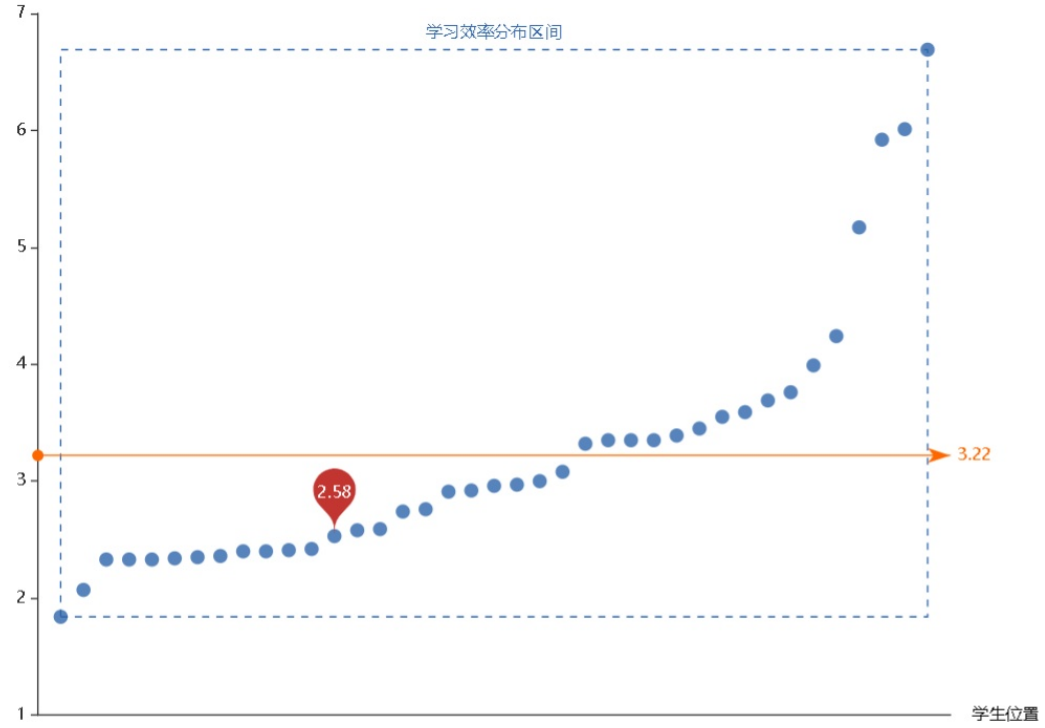
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-05-07 10:24	13	8	2024-05-07 10:41	17分 46秒	否	100/100	14.28/14.28	14.28
2	栈空判断	2024-05-07 10:43	23	7	2024-05-07 10:48	5分 17秒	否	100/100	14.28/14.28	14.28
3	栈满判断	2024-05-07 10:49	26	4	2024-05-07 10:52	2分 44秒	否	100/100	14.28/14.28	14.28
4	压栈	2024-05-07 10:52	35	5	2024-05-07 10:59	7分 2秒	否	100/100	11.28/14.28	11.28
5	弹栈	2024-05-07 11:00	40	2	2024-05-07 11:05	5分 6秒	否	100/100	14.28/14.28	14.28
6	取栈顶元素	2024-05-07	45	6	2024-05-07	4分 0秒	否	100/10	14.28/14.2	14.28

		11:05			11:09			0	8	
7	利用栈实现数进制转换	2024-05-07 11:10	110	7	2024-05-07 11:46	31分 29秒	否	300/300	14.32/14.32	14.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名 黄心媛
学号 2340850073
学习效率 2.58
当前排名 26

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

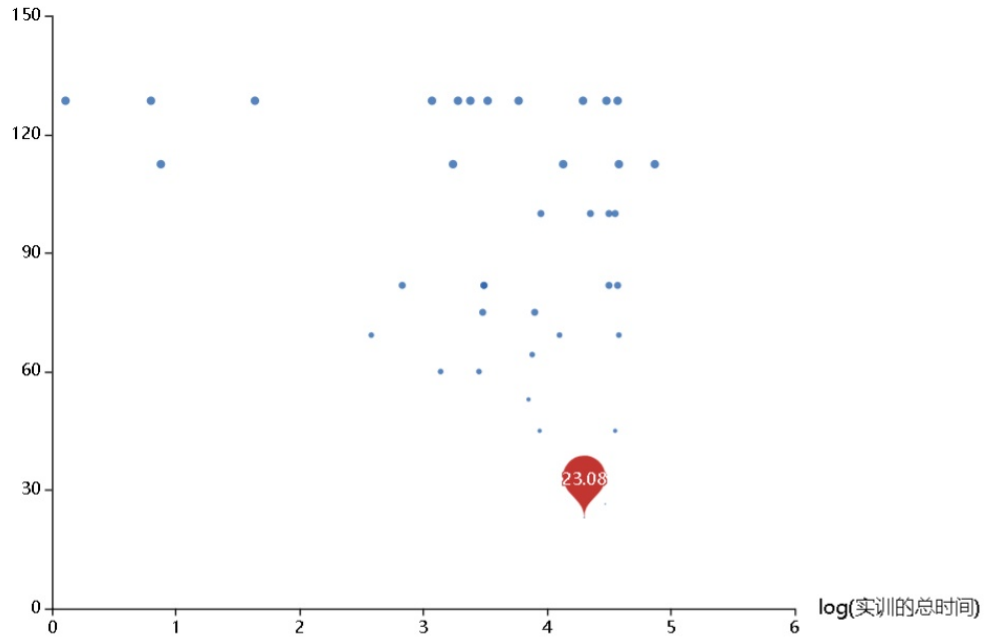
能力值

能力值（实训获得经验值/实训评测次数）

171

姓名 黄心媛
学号 2340850073

能力 23.08



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：14.28

学员代码 ⓘ　　代码修改行数 13　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
  int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
  int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
  DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
  //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有2组测试结果不匹配	0.03	3659.94	2024-05-07 10:31	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	3659.94	2024-05-07 10:32	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0.01	182.85	2024-05-07 10:33	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0.01	115.92	2024-05-07 10:37	查看代码
5	评测通过	0.02	497.82	2024-05-07 10:41	查看代码
6	评测通过	0.01	497.82	2024-05-07 10:42	查看代码
7	评测通过	0.04	3659.94	2024-05-07 10:42	查看代码

第2关 栈空判断

调分：14.28

学员代码 ⓘ　　代码修改行数 23　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.03	320.7	2024-05-07 10:45	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	331.88	2024-05-07 10:46	查看代码
3	共有2组测试集，其中有1组测试结果不匹配	0.02	328.24	2024-05-07 10:46	查看代码
4	共有2组测试集，其中有1组测试结果不匹配	0.01	347.2	2024-05-07 10:47	查看代码
5	评测通过	0.02	93.7	2024-05-07 10:48	查看代码
6	共有2组测试集，其中有1组测试结果不匹配	0.03	46.46	2024-05-07 10:48	查看代码
7	评测通过	0.02	354.24	2024-05-07 10:49	查看代码

第3关 栈满判断

调分：14.28

学员代码 ⓘ　　代码修改行数 26　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	57.66	2024-05-07 10:50	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	45.74	2024-05-07 10:51	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0.01	51.16	2024-05-07 10:52	查看代码
4	评测通过	0.01	184.71	2024-05-07 10:52	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 压栈

调分: 11.28

学员代码 代码修改行数 35 src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
  int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
  int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可 整型 curNum
  DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
 //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有1组测试结果不匹配	0.02	97.52	2024-05-07 10:54	查看代码
2	共有2组测试集，其中有1组测试结果不匹配	0.02	2439.5	2024-05-07 10:56	查看代码
3	共有2组测试集，其中有1组测试结果不匹配	0.02	322.74	2024-05-07 10:58	查看代码
4	共有2组测试集，其中有1组测试结果不匹配	0.01	322.74	2024-05-07 10:58	查看代码
5	评测通过	0.01	322.74	2024-05-07 10:59	查看代码

代码评价 一键AI生成 编辑

176

暂无评价~

第5关 弹栈

调分：14.28

学员代码 ⓘ　　代码修改行数 40　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	343.78	2024-05-07 11:05	查看代码
2	评测通过	0	324.74	2024-05-07 11:05	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 取栈顶元素

调分：14.28

学员代码 ⓘ　　代码修改行数 45　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有1组测试结果不匹配	0.02	303.82	2024-05-07 11:06	查看代码
2	共有2组测试集，其中有1组测试结果不匹配	0.02	343.78	2024-05-07 11:09	查看代码
3	评测通过	0.01	319.92	2024-05-07 11:09	查看代码
4	共有2组测试集，其中有1组测试结果不匹配	0.02	303.82	2024-05-07 11:10	查看代码
5	评测通过	0.01	319.92	2024-05-07 11:10	查看代码
6	评测通过	0.01	2439.82	2024-05-07 11:14	查看代码

代码评价　　一键AI生成　编辑

第7关 利用栈实现数进制转换

调分：14.32

学员代码 ⓘ　　代码修改行数 110　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.03	3057.6	2024-05-07 11:18	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.02	441.87	2024-05-07 11:18	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0.03	441.87	2024-05-07 11:18	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0.01	489.33	2024-05-07 11:19	查看代码
5	共有3组测试集，其中有3组测试结果不匹配	0.02	445.02	2024-05-07 11:22	查看代码
6	共有3组测试集，其中有1组测试结果不匹配	0	3659.61	2024-05-07 11:40	查看代码
7	评测通过	0	497.85	2024-05-07 11:46	查看代码

总体评价

点评作业



周钰欣 按时通关

学号：2340850045　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-07 11:34	页面停留时长	1时 30分 8秒	11	- -	- -	89.0/100.0	良好

阶段成绩

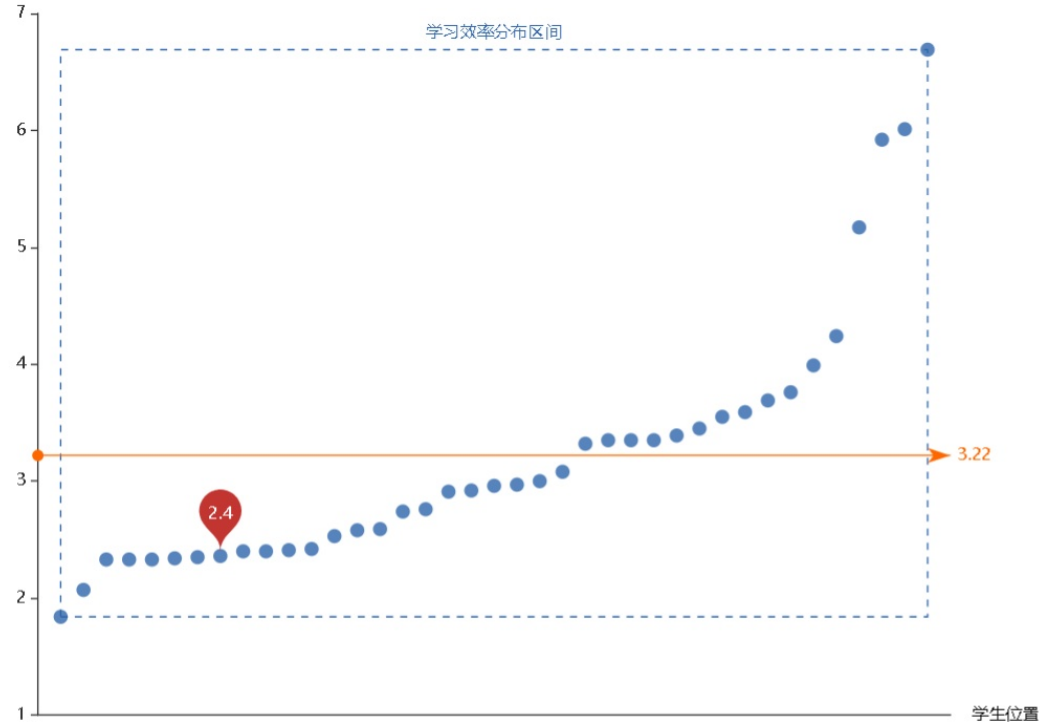
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-04-30 11:45	13	1	2024-05-07 10:31	27分 44秒	否	100/100	14.28/14.28	14.28
2	栈空判断	2024-05-07 10:31	19	3	2024-05-07 10:35	4分 38秒	否	100/100	14.28/14.28	14.28
3	栈满判断	2024-05-07 10:35	23	1	2024-05-07 10:38	2分 17秒	否	100/100	14.28/14.28	14.28
4	压栈	2024-05-07 10:38	31	1	2024-05-07 10:42	4分 3秒	否	100/100	14.28/14.28	14.28
5	弹栈	2024-05-07 10:42	39	1	2024-05-07 10:47	5分 20秒	否	100/100	11.28/14.28	11.28
6	取栈顶元素	2024-05-07	51	1	2024-05-07	27分 44	否	100/10	10.28/14.2	10.28

		10:47			11:15	秒		0	8	
7	利用栈实现数进制转换	2024-05-07 11:15	111	3	2024-05-07 11:34	18分 22秒	否	300/300	10.32/14.32	10.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	周钰欣
学号	2340850045
学习效率	2.4
当前排名	31

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

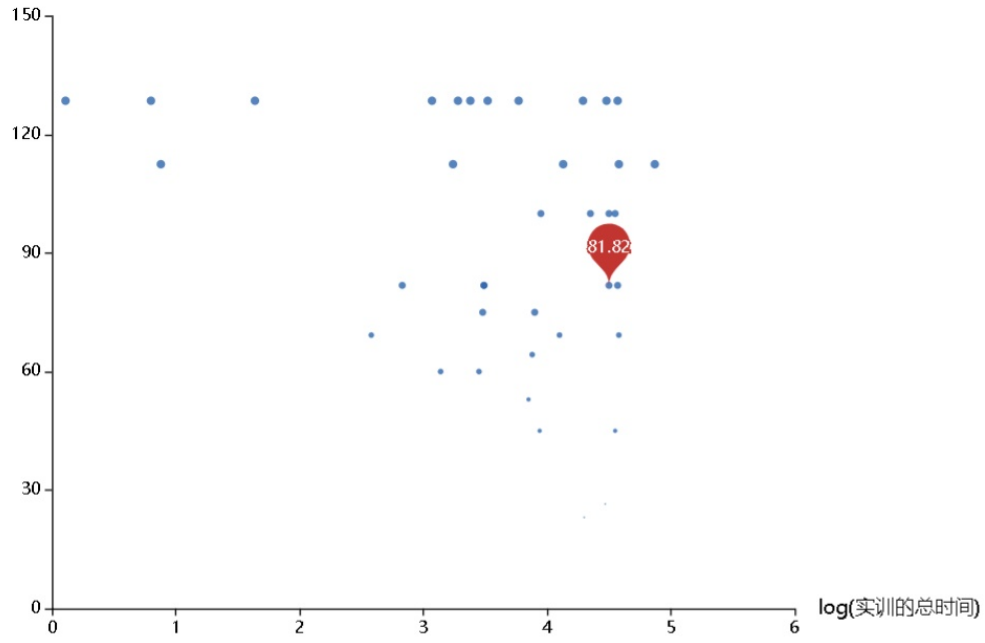
能力值

能力值（实训获得经验值/实训评测次数）

181

姓名	周钰欣
学号	2340850045

能力 81.82



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：14.28

学员代码 ⓘ　　代码修改行数 13　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	455.04	2024-05-07 10:31	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 栈空判断

调分：14.28

学员代码 ⓘ　　代码修改行数 19　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	140.28	2024-05-07 10:33	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	140.28	2024-05-07 10:34	查看代码
3	评测通过	0	343.64	2024-05-07 10:35	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 栈满判断

调分：14.28

学员代码 ⓘ　　代码修改行数 23　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	519.28	2024-05-07 10:38	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 压栈

调分：14.28

学员代码 ⓘ　　代码修改行数 31　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	336.04	2024-05-07 10:42	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 弹栈

调分: 11.28

学员代码 ⓘ　　代码修改行数 39　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	345.4	2024-05-07 10:47	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 取栈顶元素

调分：10.28

学员代码 ⓘ　　代码修改行数 51　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	597.94	2024-05-07 11:15	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第7关 利用栈实现数进制转换

调分: 10.32

学员代码 ⓘ　　代码修改行数 111　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.01	484.29	2024-05-07 11:30	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0.01	527.31	2024-05-07 11:30	查看代码
3	评测通过	0.02	469.56	2024-05-07 11:34	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



马文韬 按时通关

学号：2340850067　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 21:03	页面停留时长	21分 27秒	7	- -	- -	85.0/100.0	良好

阶段成绩

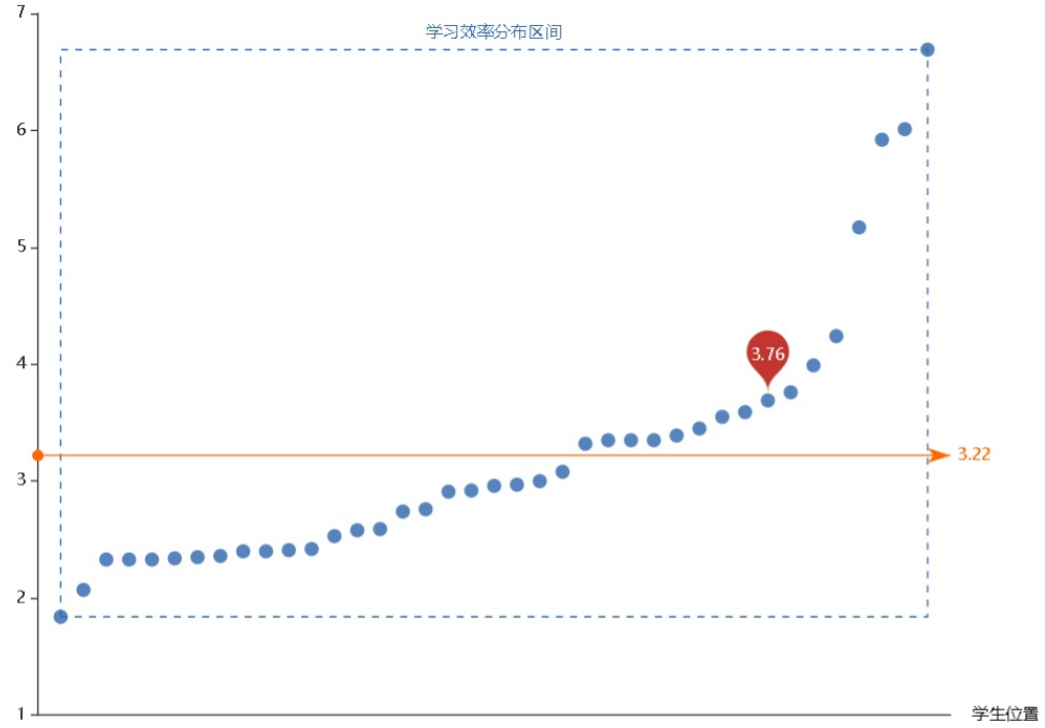
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-05-21 20:41	119	1	2024-05-21 21:02	20分 55秒	否	100/100	14.28/14.28	14.28
2	栈空判断	2024-05-21 21:02	119	1	2024-05-21 21:02	5秒	否	100/100	11.28/14.28	11.28
3	栈满判断	2024-05-21 21:02	119	1	2024-05-21 21:02	7秒	否	100/100	14.28/14.28	14.28
4	压栈	2024-05-21 21:02	119	1	2024-05-21 21:02	4秒	否	100/100	12.28/14.28	12.28
5	弹栈	2024-05-21 21:02	119	1	2024-05-21 21:02	4秒	否	100/100	9.28/14.28	9.28
6	取栈顶元素	2024-05-21	119	1	190 2024-05-21	8秒	否	100/10	14.28/14.2	14.28

		21:03			21:03			0	8	
7	利用栈实现数进制转换	2024-05-21 21:03	119	1	2024-05-21 21:03	4秒	否	300/300	9.32/14.32	9.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名 马文韬
学号 2340850067
学习效率 3.76
当前排名 7

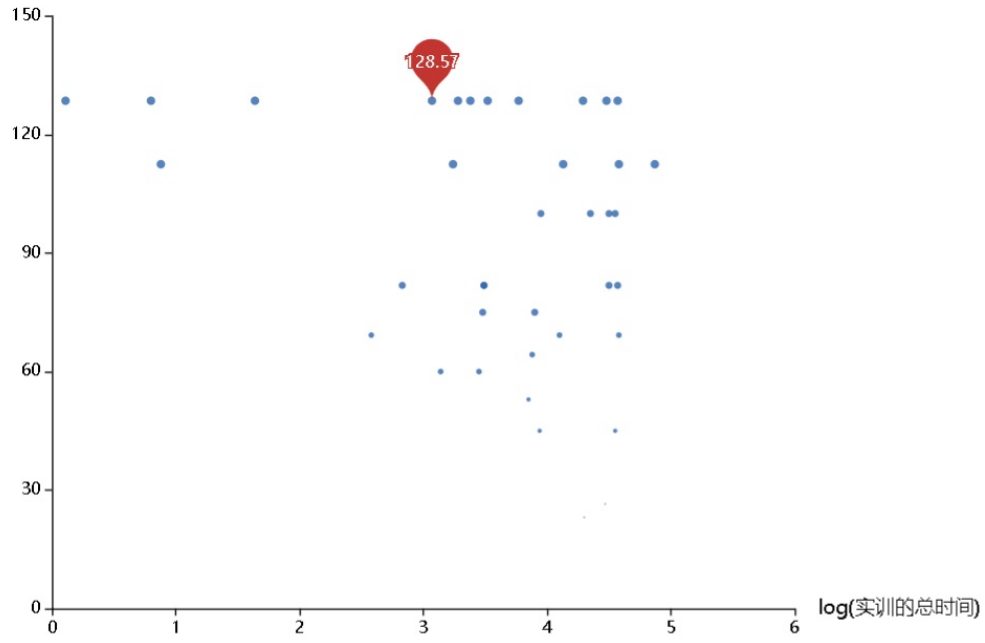
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名 马文韬
学号 2340850067

能力 128.57



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：14.28

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	199.44	2024-05-21 21:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 栈空判断

调分: 11.28

学员代码 | 代码修改行数 119 | src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	359.82	2024-05-21 21:02	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 栈满判断

调分：14.28

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
  int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
  int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
  DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
  //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1219.76	2024-05-21 21:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 压栈

调分：12.28

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	2439.52	2024-05-21 21:02	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第5关 弹栈

调分: 9.28

学员代码 | 代码修改行数 119 | src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	329.56	2024-05-21 21:02	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 取栈顶元素

调分：14.28

学员代码 ⓘ　　代码修改行数 119　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	2439.52	2024-05-21 21:03	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第7关 利用栈实现数进制转换

调分: 9.32

学员代码 | 代码修改行数 119 | src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	3659.28	2024-05-21 21:03	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



叶思含 按时通关

学号：2340850051　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-06 21:00	页面停留时长	1时 29分 51秒	9	- -	- -	79.0/100.0	良好

阶段成绩

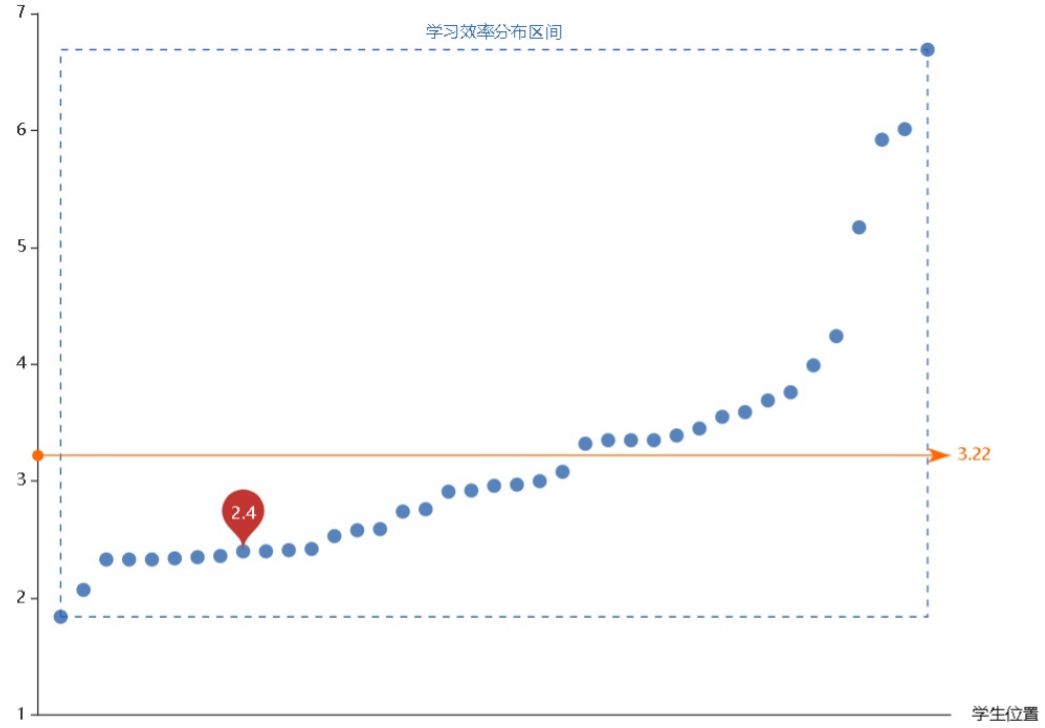
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-05-06 18:58	17	2	2024-05-06 19:31	10分 35秒	否	100/100	9.28/14.28	9.28
2	栈空判断	2024-05-06 19:31	26	1	2024-05-06 19:37	6分 4秒	否	100/100	14.28/14.28	14.28
3	栈满判断	2024-05-06 19:39	31	1	2024-05-06 19:43	3分 52秒	否	100/100	9.28/14.28	9.28
4	压栈	2024-05-06 19:43	40	1	2024-05-06 19:49	5分 47秒	否	100/100	11.28/14.28	11.28
5	弹栈	2024-05-06 19:49	45	1	2024-05-06 19:58	8分 47秒	否	100/100	11.28/14.28	11.28
6	取栈顶元素	2024-05-06	50	2	200 2024-05-06	1分 2秒	否	100/10	9.28/14.28	9.28

		19:58			19:59			0		
7	利用栈实现数进制转换	2024-05-06 19:59	79	1	2024-05-06 21:00	53分 44秒	否	300/300	14.32/14.32	14.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名 叶思含
学号 2340850051
学习效率 2.4
当前排名 30

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

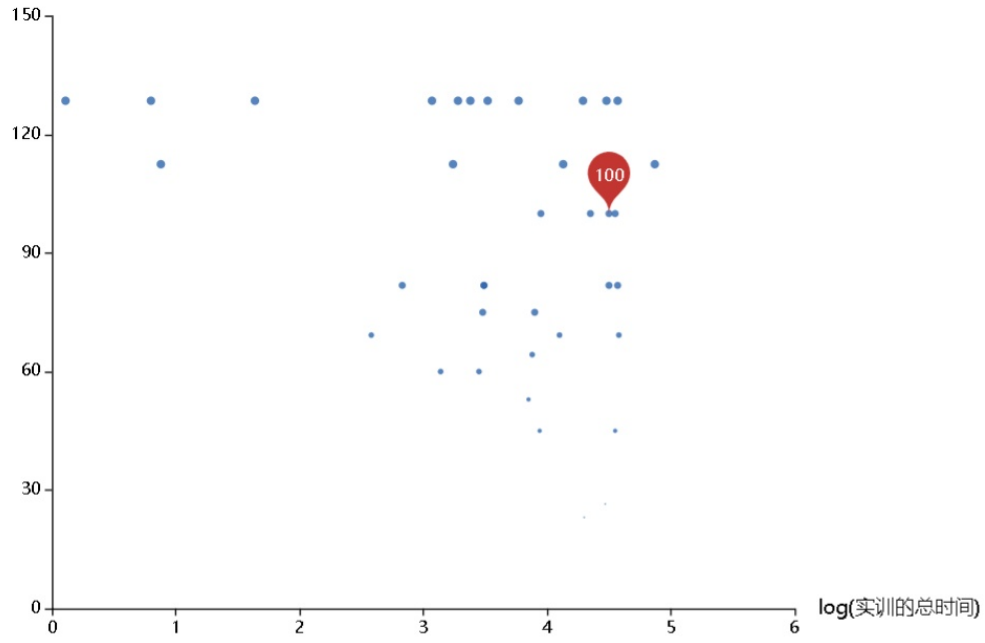
能力值

能力值（实训获得经验值/实训评测次数）

201

姓名 叶思含
学号 2340850051

能力 100



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：9.28

学员代码 ⓘ　　代码修改行数 17　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有1组测试结果不匹配	0.03	206.19	2024-05-06 19:29	查看代码
2	评测通过	0.03	1868.52	2024-05-06 19:31	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 栈空判断

调分：14.28

学员代码 ⓘ　　代码修改行数 26　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	1518.78	2024-05-06 19:37	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第3关 栈满判断

调分：9.28

学员代码 ⓘ　　代码修改行数 31　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	58.09	2024-05-06 19:43	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 压栈

调分：11.28

学员代码 ⓘ　　代码修改行数 40　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	2440.04	2024-05-06 19:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 弹栈

调分：11.28

学员代码 ⓘ　　代码修改行数 45　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1738.5	2024-05-06 19:58	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第6关 取栈顶元素

调分：9.28

学员代码 ⓘ　　代码修改行数 50　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	2106.88	2024-05-06 19:59	查看代码
2	评测通过	0	2014.54	2024-05-06 20:02	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第7关 利用栈实现数进制转换

调分：14.32

学员代码 ⓘ　　代码修改行数 79　　src/seqlist.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    //若m=0，则返回NULL
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	796.44	2024-05-06 21:00	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



张丽芳 按时通关

学号：2340850068　　分班：未分班

截止前完成关卡：7/7　　最新完成关卡：7/7　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-08 20:12	页面停留时长	1时 34分 19秒	20	- -	- -	77.0/100.0	良好

阶段成绩

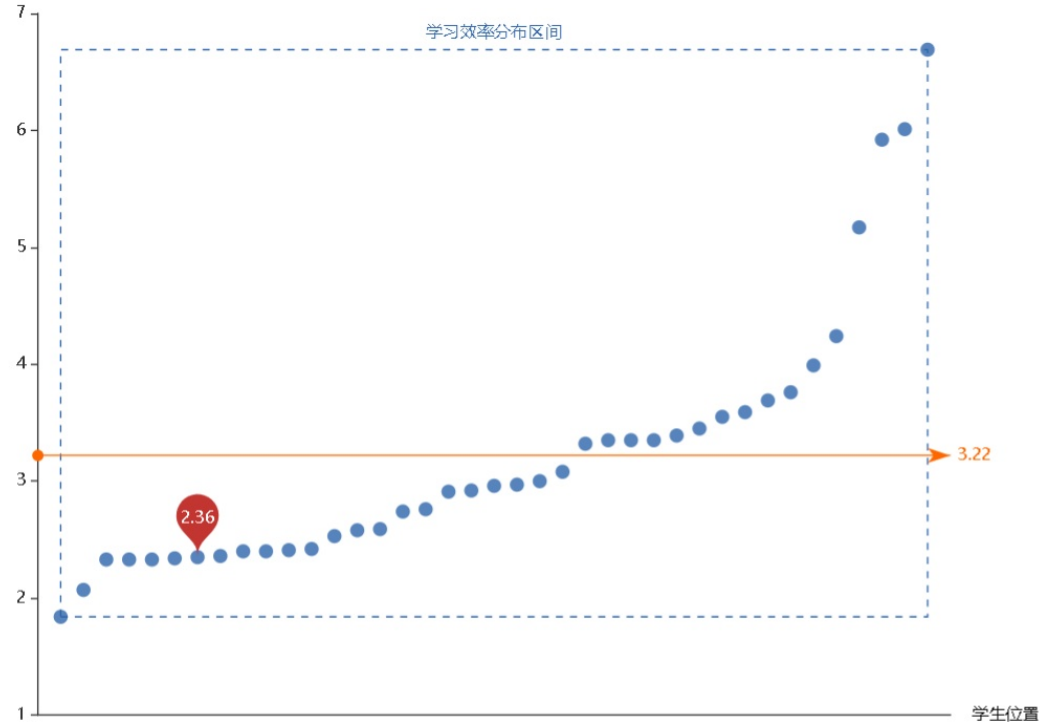
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建空的顺序栈	2024-05-07 10:25	15	1	2024-05-07 10:47	21分 22秒	否	100/100	10.28/14.28	10.28
2	栈空判断	2024-05-07 10:26	22	2	2024-05-07 10:53	5分 14秒	否	100/100	10.28/14.28	10.28
3	栈满判断	2024-05-07 10:26	24	3	2024-05-07 10:56	3分 26秒	否	100/100	12.28/14.28	12.28
4	压栈	2024-05-07 10:26	32	6	2024-05-07 11:13	16分 10秒	否	100/100	10.28/14.28	10.28
5	弹栈	2024-05-07 10:26	41	2	2024-05-07 11:22	9分 50秒	否	100/100	9.28/14.28	9.28
6	取栈顶元素	2024-05-07	52	1	2024-05-07	6分 55秒	否	100/10	10.28/14.2	10.28

		10:26			11:29			0	8	
7	利用栈实现数进制转换	2024-05-07 10:26	114	5	2024-05-08 20:12	31分 22秒	否	300/300	14.32/14.32	14.32

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名 张丽芳
学号 2340850068
学习效率 2.36
当前排名 32

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

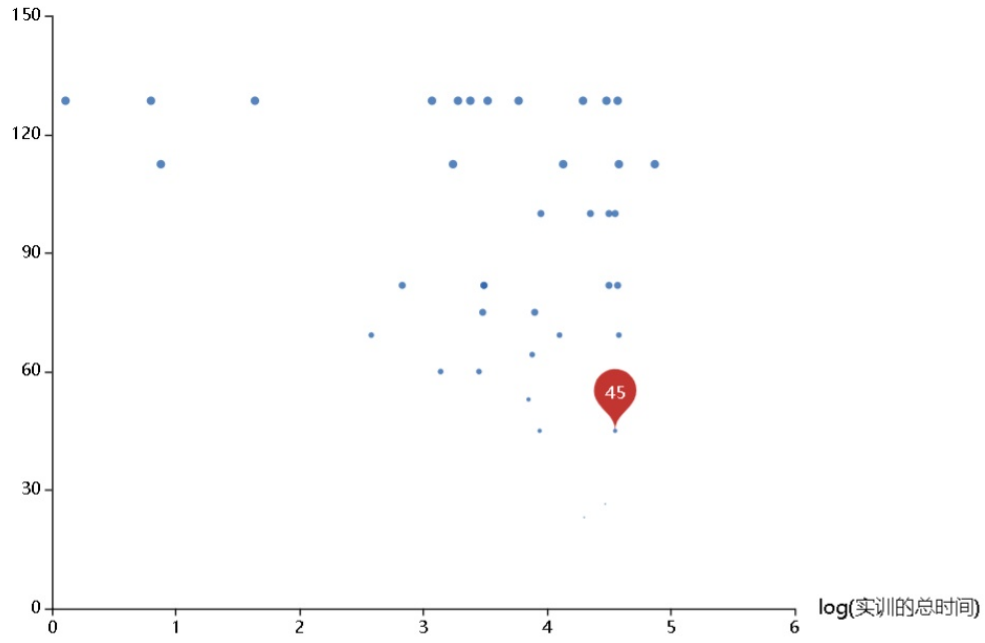
能力值

能力值（实训获得经验值/实训评测次数）

211

姓名 张丽芳
学号 2340850068

能力 45



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 创建空的顺序栈

调分：10.28

学员代码 ⓘ　　代码修改行数 15　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	3659.22	2024-05-07 10:47	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第2关 栈空判断

调分：10.28

学员代码 ⓘ　　代码修改行数 22　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	298.14	2024-05-07 10:51	查看代码
2	评测通过	0.01	312.58	2024-05-07 10:53	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 栈满判断

调分: 12.28

学员代码 代码修改行数 24 src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
  int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
  int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可 整型 curNum
  DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
  PseqStack p=(PseqStack*)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	157.89	2024-05-07 10:55	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	1219.6	2024-05-07 10:56	查看代码
3	评测通过	0	160.45	2024-05-07 10:56	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第4关 压栈

调分：10.28

学员代码 ⓘ　　代码修改行数 32　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	1737.66	2024-05-07 11:10	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	1822.14	2024-05-07 11:11	查看代码
3	共有2组测试集，其中有1组测试结果不匹配	0.02	335.5	2024-05-07 11:11	查看代码
4	共有2组测试集，其中有1组测试结果不匹配	0.02	1737.66	2024-05-07 11:11	查看代码
5	共有2组测试集，其中有1组测试结果不匹配	0.01	1749.32	2024-05-07 11:12	查看代码
6	评测通过	0.03	310.6	2024-05-07 11:13	查看代码

代码评价　　一键AI生成　编辑

第5关 弹栈

调分: 9.28

学员代码 代码修改行数 41 src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	0	2024-05-07 11:17	查看代码
2	评测通过	0.02	1797.84	2024-05-07 11:22	查看代码

代码评价

一键AI生成 编辑

暂无评价~

调分记录

第6关 取栈顶元素

调分: 10.28

学员代码 代码修改行数 52 src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	313.32	2024-05-07 11:29	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第7关 利用栈实现数进制转换

调分：14.32

学员代码 ⓘ　　代码修改行数 114　　src/seqlist.h

复制代码

```
#include <stdio.h>

/*此处是顺序栈数据结构定义*/
typedef int DataType;
struct seqStack
{//有3个数据成员
    int MAXNUM;//用于记录顺序栈中能存放的最大元素个数的 整型 MAXNUM
    int top;//用于存放顺序栈的栈顶位置，初始化为0或-1 均可  整型  curNum
    DataType *element;//用于存放顺序栈数据元素的连续空间的起始地址
};

typedef struct seqStack *PseqStack;
//第一关
PseqStack createNullStack_seq(int m)
{//此处填写代码，创建一个空的顺序栈，能存放的最大元素个数为 m,栈顶top设置为0
    PseqStack p=(PseqStack)malloc(sizeof(struct seqStack));
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	75.42	2024-05-07 11:54	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	213.21	2024-05-07 11:56	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0	476.31	2024-05-07 11:57	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0	2815.47	2024-05-08 20:11	查看代码
5	评测通过	0	456.42	2024-05-08 20:12	查看代码

代码评价

一键AI生成　编辑

219

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 11:18	页面停留时长	1时 17分 15秒	18	- -	- -	92.0/100.0	优秀

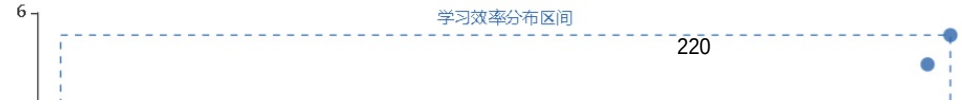
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-07 11:54	24	8	2024-05-14 11:34	17分 50秒	否	100/100	25.00/25	25
2	二叉树基本操作	2024-05-14 11:32	70	7	2024-05-21 10:58	40分 39秒	否	100/100	25.00/25	25
3	二叉树的层序遍历	2024-05-21 10:36	77	1	2024-05-21 11:01	2分 28秒	否	100/100	22.00/25	22
4	二叉树的先序遍历	2024-05-21 11:01	96	2	2024-05-21 11:18	16分 18秒	否	100/100	20.00/25	20

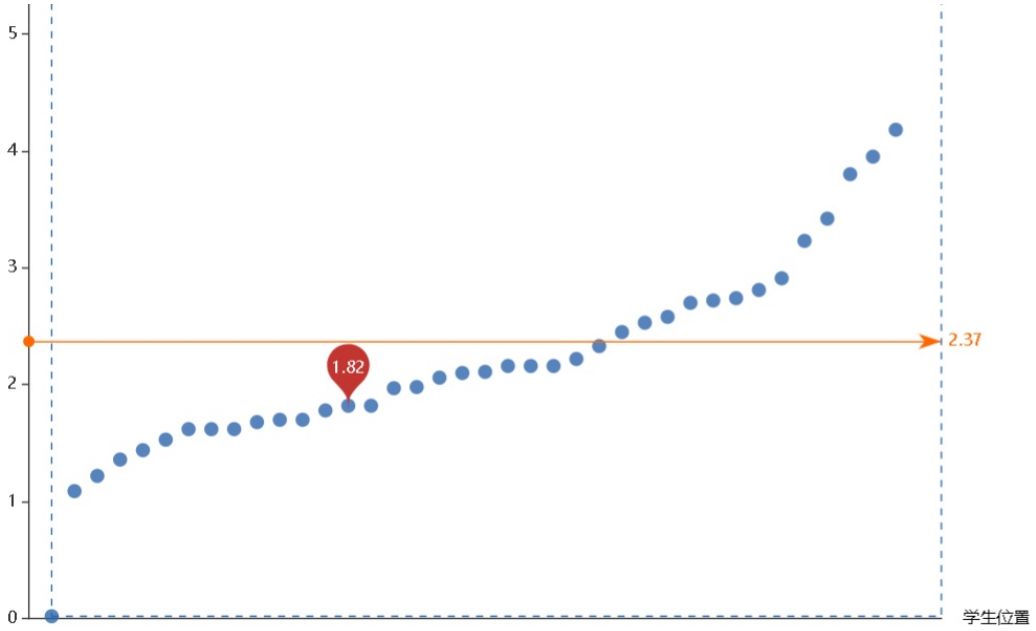
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	李雯君
学号	2340850043
学习效率	1.82
当前排名	26



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

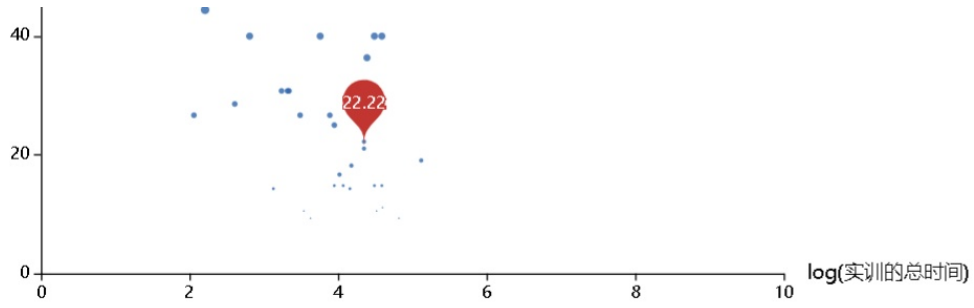
能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	22.22



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分：25

学员代码 ⓘ　　代码修改行数 24　　seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
*/
struct seqTree
{
  int n;
  char *root;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i!=T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2440.96	2024-05-14 11:26	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	302	2024-05-14 11:32	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	1828.42	2024-05-14 11:33	查看代码
4	评测通过	0	320.12	2024-05-14 11:34	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0.02	319.36	2024-05-21 10:44	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0.02	319.36	2024-05-21 10:45	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0	309.06	2024-05-21 10:49	查看代码

第2关 二叉树基本操作

调分：25

学员代码 代码修改行数 70 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
*/
struct seqTree
{
  int n;
  char *root;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i!=T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.04	107.66	2024-05-14 11:57	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	364.48	2024-05-21 10:35	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	336.78	2024-05-21 10:54	查看代码
4	共有2组测试集，其中有1组测试结果不匹配	0.02	477	2024-05-21 10:55	查看代码
5	共有2组测试集，其中有1组测试结果不匹配	0.02	339.28	2024-05-21 10:56	查看代码
6	共有2组测试集，其中有1组测试结果不匹配	0.03	380	2024-05-21 10:57	查看代码
7	评测通过	0.01	322.22	2024-05-21 10:58	查看代码

第3关 二叉树的层序遍历

调分：22

学员代码 ⓘ　　代码修改行数 77　　seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
*/
struct seqTree
{
    int n;
    char *root;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
    int i = 0;
    while(i!=T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	220.82	2024-05-21 11:01	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 二叉树的先序遍历

调分: 20

学员代码 代码修改行数 96 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
*/
struct seqTree
{
    int n;
    char *root;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
    int i = 0;
    while(i!=T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	316.5	2024-05-21 11:05	查看代码
2	评测通过	0.02	1810.08	2024-05-21 11:18	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-22 22:25	页面停留时长	1时 38分 24秒	10	- -	- -	93.0/100.0	优秀

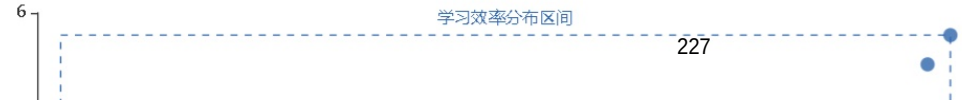
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-21 10:43	23	1	2024-05-21 11:10	26分 58秒	否	100/100	25.00/25	25
2	二叉树基本操作	2024-05-21 11:10	62	6	2024-05-22 16:40	49分 21秒	否	100/100	24.00/25	24
3	二叉树的层序遍历	2024-05-22 16:43	66	1	2024-05-22 16:47	3分 55秒	否	100/100	24.00/25	24
4	二叉树的先序遍历	2024-05-22 16:47	75	2	2024-05-22 22:25	18分 10秒	否	100/100	20.00/25	20

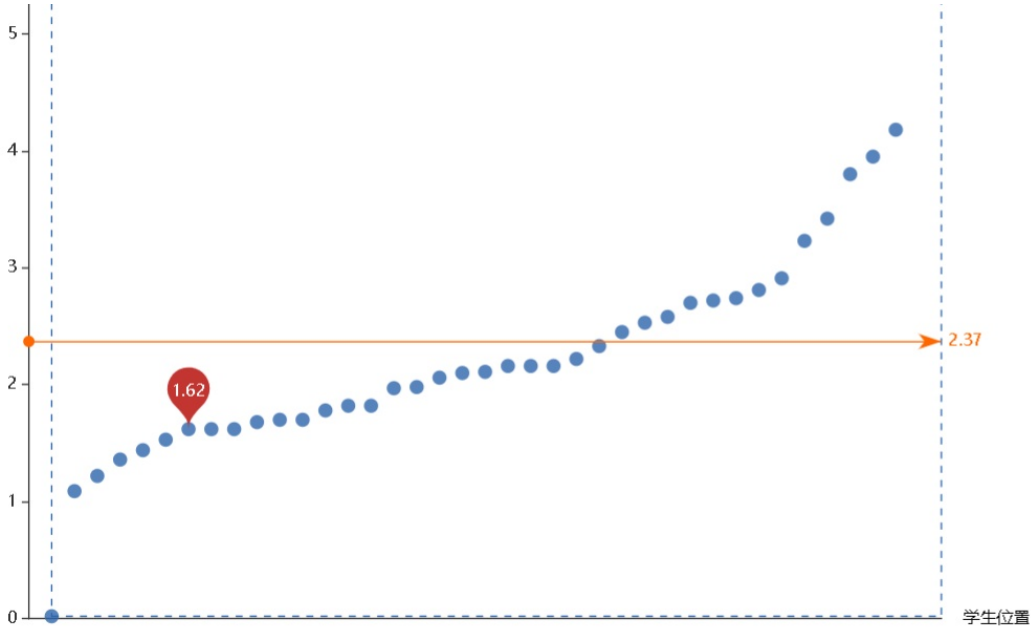
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	黄心媛
学号	2340850073
学习效率	1.62
当前排名	33



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

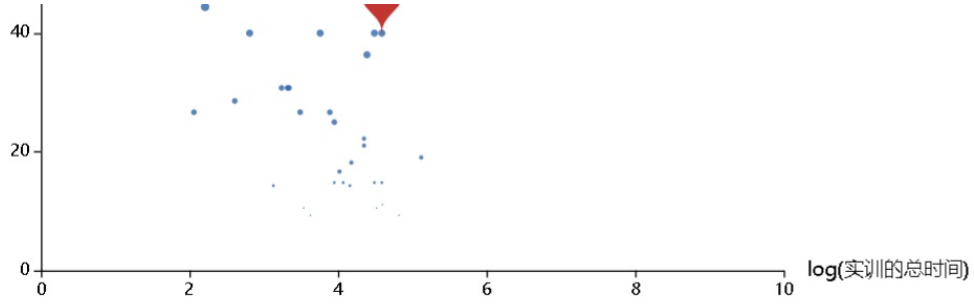
能力值

能力值（实训获得经验值/实训评测次数）

姓名	黄心媛
学号	2340850073
能力	40



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分: 25

学员代码 代码修改行数 23 seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{
    DataType *data;
    int nodenum;
    int maxnum;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	332	2024-05-21 11:10	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 二叉树基本操作

调分：24

学员代码 ⓘ　　代码修改行数 62　　seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{
    DataType *data;
    int nodenum;
    int maxnum;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	412.28	2024-05-21 11:43	查看代码
2	共有2组测试集，其中有1组测试结果不匹配	0.03	337.98	2024-05-21 11:47	查看代码
3	评测通过	0.04	2033.84	2024-05-22 16:40	查看代码
4	评测通过	0.02	191.9	2024-05-22 16:40	查看代码
5	共有2组测试集，其中有1组测试结果不匹配	0.01	362.7	2024-05-22 16:40	查看代码
6	评测通过	0.02	2033.84	2024-05-22 16:41	查看代码

代码评价　　一键AI生成　编辑

第3关 二叉树的层序遍历

调分：24

学员代码 ⓘ　　代码修改行数 66　　seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
/*
定义 顺序 存储 结构
 */
typedef char DataType;
struct seqTree
{
    DataType *data;
    int nodenum;
    int maxnum;
};

/*辅助 功能：逐个 输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	181.35	2024-05-22 16:47	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 二叉树的先序遍历

调分：20

学员代码 代码修改行数 75 seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{
    DataType *data;
    int nodenum;
    int maxnum;
};

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	134.08	2024-05-22 22:23	查看代码
2	评测通过	0.02	123.68	2024-05-22 22:25	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



张佳慧 按时通关

学号：2340850044　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 11:39	页面停留时长	2时 4分 29秒	43	- -	- -	87.0/100.0	良好

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-07 11:08	25	27	2024-05-21 11:27	1时 21分 34秒	否	100/100	25.00/25	25
2	二叉树基本操作	2024-05-07 11:25	52	6	2024-05-21 11:33	15分 56秒	否	100/100	20.00/25	20
3	二叉树的层序遍历	2024-05-14 11:14	58	6	2024-05-21 11:34	7分 56秒	否	100/100	22.00/25	22
4	二叉树的先序遍历	2024-05-14 11:49	71	4	2024-05-21 11:39	19分 3秒	否	100/100	20.00/25	20

图形统计

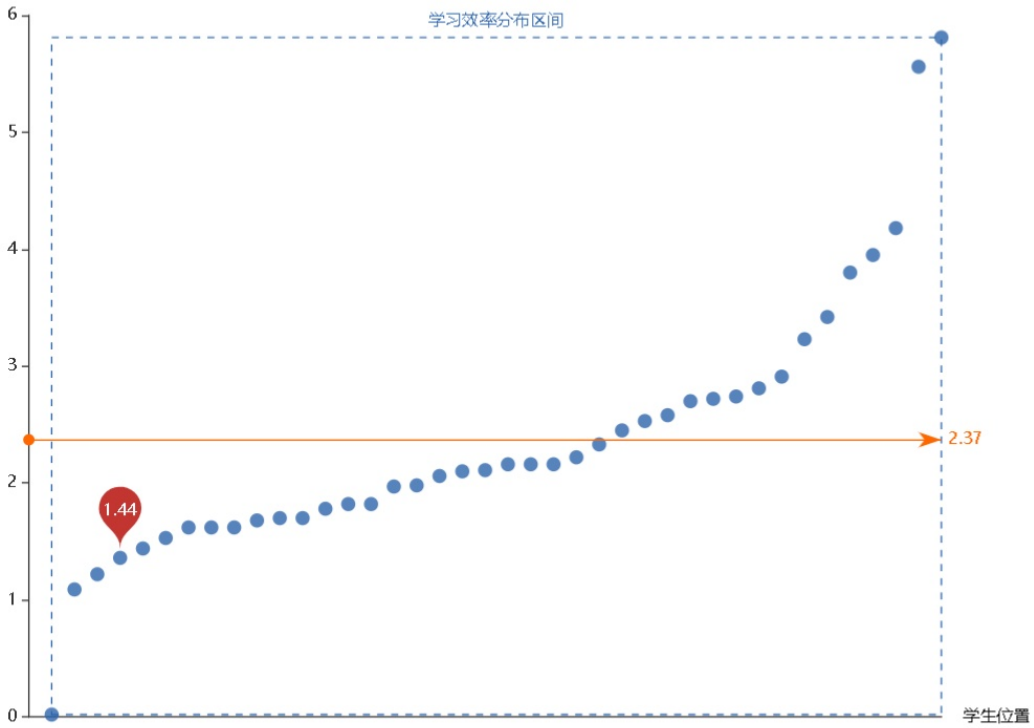
学习效率

学习效率=log(实训总得分/实训总耗时)

234

姓名 张佳慧

学号 2340850044



学习效率 1.44
当前排名 36

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

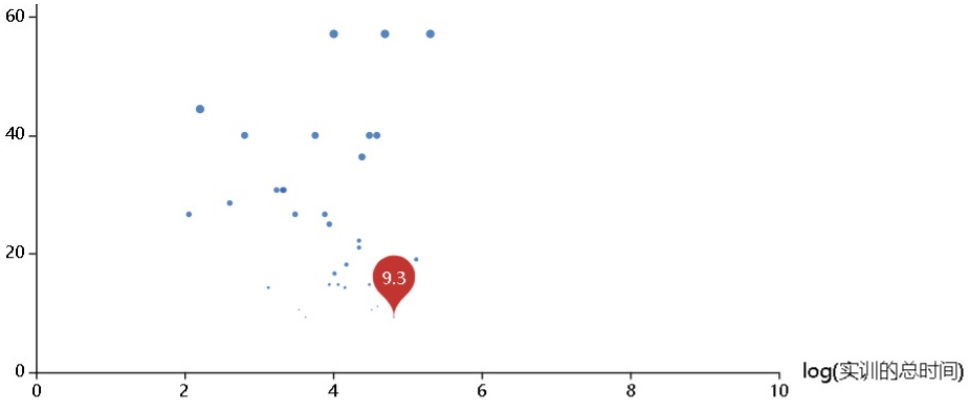
能力值

能力值（实训获得经验值/实训评测次数）



姓名 张佳慧
学号 2340850044
能力 9.3

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分：25

学员代码 ⓘ　　代码修改行数 25　　seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
 /*
 定义 顺序 存储 结构
  */
struct seqTree
{
  int n;
  char *root;
};

/*辅助 功能：逐个 输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i != T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2052.36	2024-05-07 11:19	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	315.36	2024-05-07 11:20	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	1085.46	2024-05-07 11:20	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	335.48	2024-05-07 11:22	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	1085.46	2024-05-07 11:23	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0	306.96	2024-05-07 11:23	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0.03	144.98	2024-05-07 11:25	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	315.18	2024-05-14 11:01	查看代码
9	共有2组测试集，其中有2组测试结果不匹配	0	757.98	2024-05-14 11:02	查看代码
10	共有2组测试集，其中有2组测试结果不匹配	0.02	311.82	2024-05-14 11:10	查看代码

< 2 3 >

代码评价

一键AI生成 编辑

暂无评价~

第2关 二叉树基本操作

调分：20

学员代码 ？ 代码修改行数 52 seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
 /*
 定义 顺序 存储 结构
  */
struct seqTree
{
  int n;
  char *root;
};

/*辅助 功能：逐个输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i != T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	321.4	2024-05-07 11:58	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2034.98	2024-05-14 11:01	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.01	121.94	2024-05-14 11:10	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	314.1	2024-05-14 11:49	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	318.38	2024-05-21 11:33	查看代码
6	评测通过	0.01	321.2	2024-05-21 11:33	查看代码

代码评价 一键AI生成 编辑

第3关 二叉树的层序遍历

调分：22

学员代码 ⓘ　　代码修改行数 58　　seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
 /*
 定义 顺序 存储 结构
  */
struct seqTree
{
  int n;
  char *root;
};

/*辅助 功能：逐个输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i != T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	155.91	2024-05-14 11:16	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	457.17	2024-05-14 11:16	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0	270.42	2024-05-14 11:16	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0	445.89	2024-05-14 11:17	查看代码
5	共有1组测试集，其中有1组测试结果不匹配	0.01	158.85	2024-05-14 11:49	查看代码
6	评测通过	0.01	1220.56	2024-05-21 11:34	查看代码

代码评价

一键AI生成　编辑

第4关 二叉树的先序遍历

调分：20

学员代码 ⓘ　　代码修改行数 71　　seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
 /*
 定义 顺序 存储 结构
  */
struct seqTree
{
  int n;
  char *root;
};

/*辅助 功能：逐个 输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
  int i = 0;
  while(i != T->n)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2439.44	2024-05-14 11:52	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	1598.22	2024-05-14 11:53	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.04	317.8	2024-05-14 11:53	查看代码
4	评测通过	0.04	102.5	2024-05-21 11:39	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



周钰欣 按时通关

学号：2340850045　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 11:23	页面停留时长	1时 28分 58秒	27	- -	- -	85.0/100.0	良好

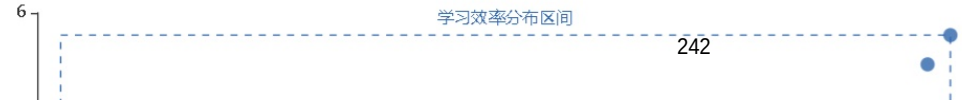
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-14 11:11	108	14	2024-05-21 10:58	24分 40秒	否	100/100	20.00/25	20
2	二叉树基本操作	2024-05-14 11:25	108	6	2024-05-21 11:22	47分 10秒	否	100/100	20.00/25	20
3	二叉树的层序遍历	2024-05-14 11:54	108	2	2024-05-21 11:22	2分 33秒	否	100/100	25.00/25	25
4	二叉树的先序遍历	2024-05-14 11:54	108	5	2024-05-21 11:23	14分 35秒	否	100/100	20.00/25	20

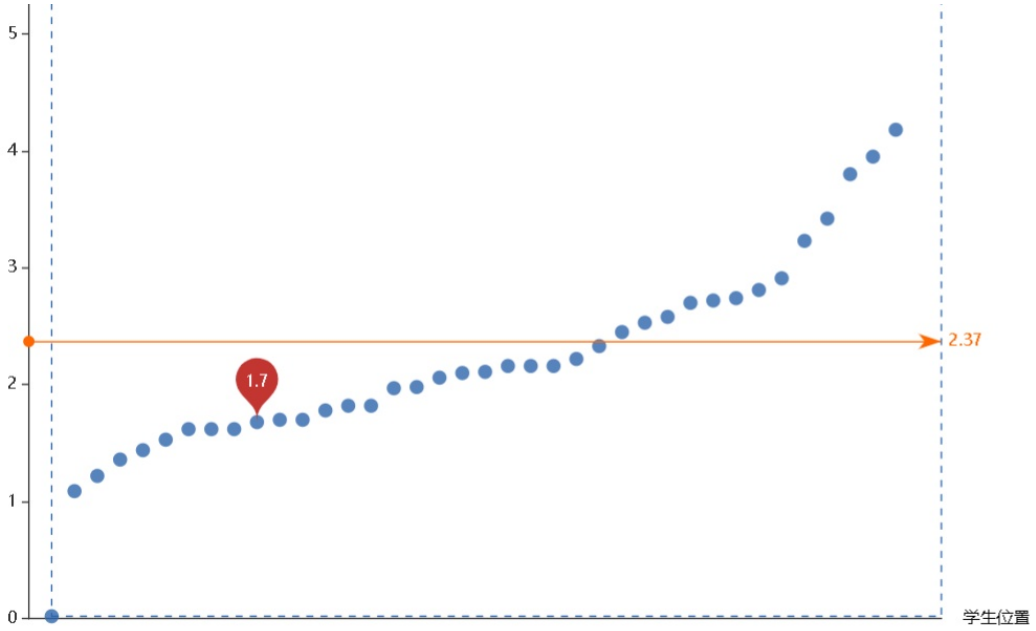
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	周钰欣
学号	2340850045
学习效率	1.7
当前排名	30



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

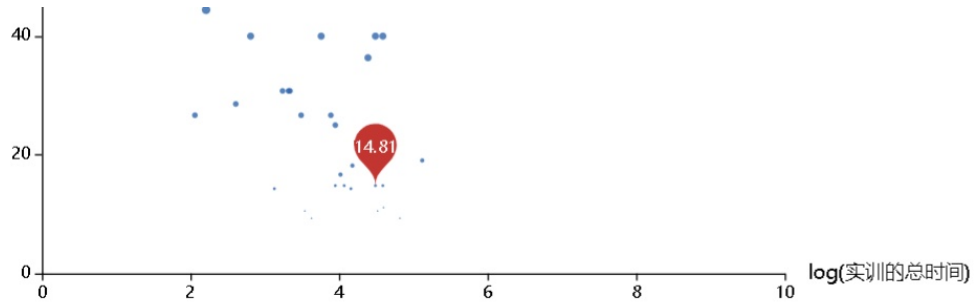
能力值

能力值（实训获得经验值/实训评测次数）

姓名	周钰欣
学号	2340850045
能力	14.81



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分：20

学员代码 代码修改行数 108 seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
/*
定义 顺序 存储 结构
 */
typedef char DataType;

struct seqTree
{
    int MAXNUM;
    int n;
    DataType* nodelist;

};


```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	312.58	2024-05-14 11:18	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.03	406	2024-05-14 11:19	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	330.4	2024-05-14 11:22	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0	1999.78	2024-05-21 10:36	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	1999.78	2024-05-21 10:37	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0	92.96	2024-05-21 10:37	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0	889.82	2024-05-21 10:40	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	1441.24	2024-05-21 10:43	查看代码
9	共有2组测试集，其中有2组测试结果不匹配	0	318.18	2024-05-21 10:53	查看代码
10	共有2组测试集，其中有2组测试结果不匹配	0	889.82	2024-05-21 10:53	查看代码

< 2 >

代码评价

一键AI生成 编辑

暂无评价~

第2关 二叉树基本操作

调分：20

学员代码 代码修改行数 108 seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
/*
定义 顺序 存储 结构
 */
typedef char DataType;

struct seqTree
{
    int MAXNUM;
    int n;
    DataType* nodelist;

};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	335.94	2024-05-14 11:35	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	330.2	2024-05-21 10:39	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.03	1441.24	2024-05-21 10:40	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	49.84	2024-05-21 10:59	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0.02	2441.08	2024-05-21 11:21	查看代码
6	评测通过	0.02	1456.9	2024-05-21 11:22	查看代码

代码评价

一键AI生成 编辑

第3关 二叉树的层序遍历

调分：25

学员代码 ⓘ　　代码修改行数 108　　seqtree.c

复制代码

```
#include <stdio.h>
#include<stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;

struct seqTree
{
    int MAXNUM;
    int n;
    DataType* nodelist;

};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	158.71	2024-05-21 10:44	查看代码
2	评测通过	0.01	1220.54	2024-05-21 11:22	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 二叉树的先序遍历

调分：20

学员代码 代码修改行数 108 seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
/*
定义 顺序 存储 结构
 */
typedef char DataType;

struct seqTree
{
    int MAXNUM;
    int n;
    DataType* nodelist;

};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2439.54	2024-05-14 11:56	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2439.86	2024-05-14 11:57	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	1998	2024-05-21 10:33	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.01	883.8	2024-05-21 10:36	查看代码
5	评测通过	0.02	2441.08	2024-05-21 11:23	查看代码

代码评价

一键AI生成 编辑

249

暂无评价~

总体评价

点评作业



徐雨秋 按时通关

学号：2340850053　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 11:18	页面停留时长	58分 29秒	27	- -	- -	79.0/100.0	良好

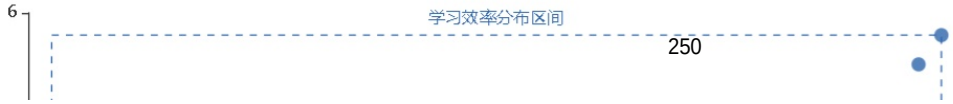
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-14 11:36	74	11	2024-05-21 10:49	30分 57秒	否	100/100	20.00/25	20
2	二叉树基本操作	2024-05-21 10:40	76	3	2024-05-21 10:49	5分 1秒	否	100/100	19.00/25	19
3	二叉树的层序遍历	2024-05-21 10:45	76	3	2024-05-21 10:50	2分 9秒	否	100/100	20.00/25	20
4	二叉树的先序遍历	2024-05-21 10:50	116	10	2024-05-21 11:18	20分 22秒	否	100/100	20.00/25	20

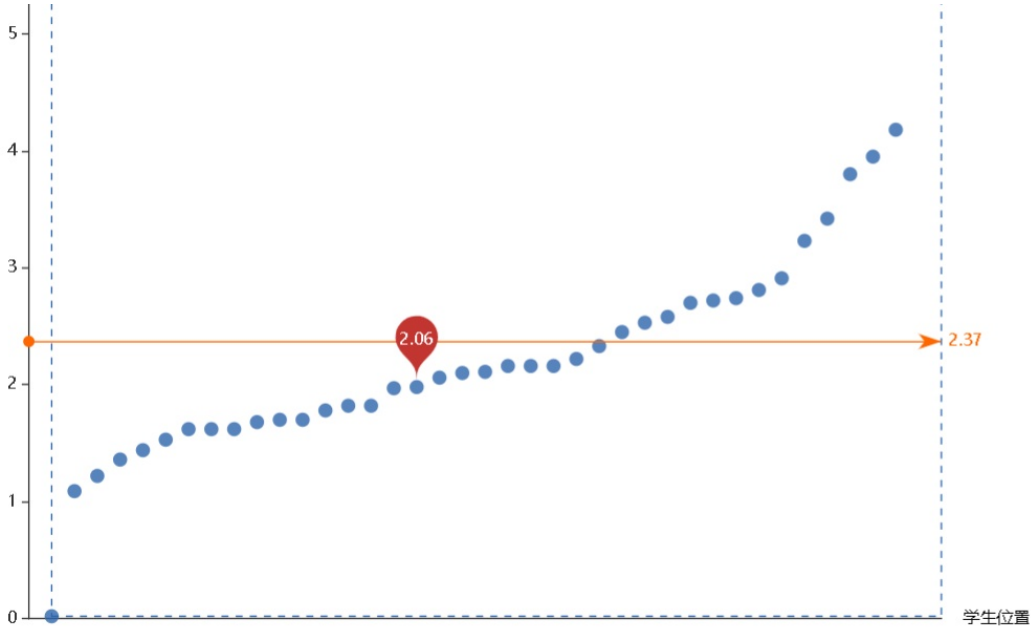
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	徐雨秋
学号	2340850053
学习效率	2.06
当前排名	23



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

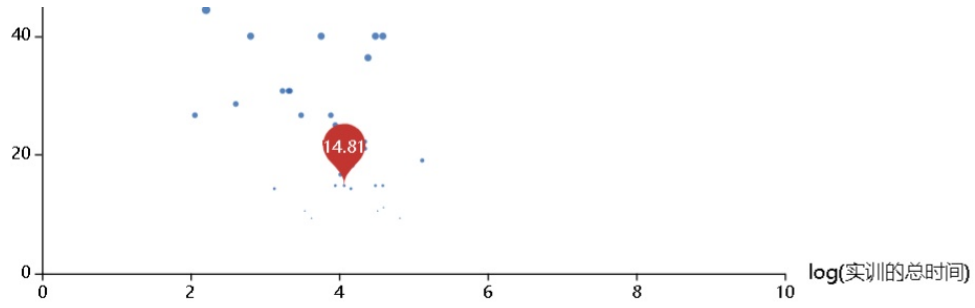
能力值

能力值（实训获得经验值/实训评测次数）

姓名	徐雨秋
学号	2340850053
能力	14.81



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分：20

学员代码 代码修改行数 74 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{ int MAXNUM;
  int n;//元素的个数
  DataType* nodelist;//数组起始地址



};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	341.24	2024-05-21 10:31	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	464.74	2024-05-21 10:32	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	308.88	2024-05-21 10:33	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	2439.64	2024-05-21 10:36	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	310.64	2024-05-21 10:37	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0.01	2212.5	2024-05-21 10:40	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0.01	2439.58	2024-05-21 10:40	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0.01	334.16	2024-05-21 10:47	查看代码
9	共有2组测试集，其中有2组测试结果不匹配	0.01	112.62	2024-05-21 10:48	查看代码
10	共有2组测试集，其中有2组测试结果不匹配	0.01	118.36	2024-05-21 10:48	查看代码

< 2 >

代码评价

一键AI生成 编辑

暂无评价~

第2关 二叉树基本操作

调分： 19

学员代码 代码修改行数 76 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{ int MAXNUM;
  int n;//元素的个数
  DataType* nodelist;//数组起始地址



};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	2439.58	2024-05-21 10:44	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.03	97.3	2024-05-21 10:49	查看代码
3	评测通过	0.02	97.3	2024-05-21 10:49	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 二叉树的层序遍历

调分：20

学员代码 ⓘ　　代码修改行数 76　　seqtree.c

复制代码

```
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
typedef char DataType;
struct seqTree
{ int MAXNUM;
  int n;//元素的个数
  DataType* nodelist;//数组起始地址



};

```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	155.25	2024-05-21 10:45	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	46	2024-05-21 10:46	查看代码
3	评测通过	0.01	1219.82	2024-05-21 10:50	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 二叉树的先序遍历

调分：20

学员代码 代码修改行数 116 seqtree.c

复制代码

```c
#include <stdio.h>
#include<stdlib.h>
/*
定义 顺序 存储 结构
 */
typedef char DataType;

struct seqTree
{
    int MAXNUM;
    int n;
    DataType* nodelist;

};


```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2439.64	2024-05-21 10:51	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	126.28	2024-05-21 10:52	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	310.5	2024-05-21 10:53	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.02	315.68	2024-05-21 10:53	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	524.38	2024-05-21 10:56	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0	134.38	2024-05-21 10:59	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0	134.38	2024-05-21 11:00	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	318.68	2024-05-21 11:03	查看代码
9	共有2组测试集，其中有2组测试结果不匹配	0	118.46	2024-05-21 11:17	查看代码
10	评测通过	0.02	337.58	2024-05-21 11:18	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



秦慧颖 按时通关

学号：2340850057　　分班：未分班

截止前完成关卡：4/4　　最新完成关卡：4/4　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-27 21:30	页面停留时长	32分 53秒	15	- -	- -	78.0/100.0	良好

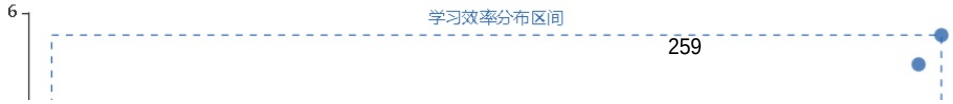
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	创建顺序结构的二叉树	2024-05-21 10:51	53	9	2024-05-27 21:30	27分 2秒	否	100/100	20.00/25	20
2	二叉树基本操作	2024-05-21 11:17	89	3	2024-05-27 21:28	5分 2秒	否	100/100	20.00/25	20
3	二叉树的层序遍历	2024-05-27 21:28	92	1	2024-05-27 21:28	22秒	否	100/100	20.00/25	20
4	二叉树的先序遍历	2024-05-27 21:28	92	2	2024-05-27 21:29	27秒	否	100/100	18.00/25	18

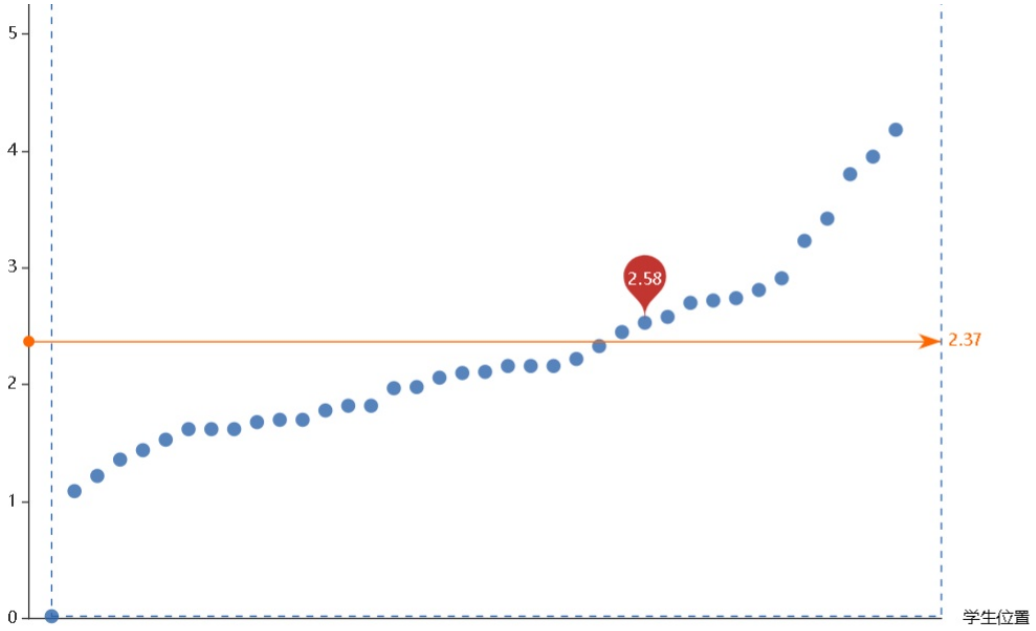
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	秦慧颖
学号	2340850057
学习效率	2.58
当前排名	13



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

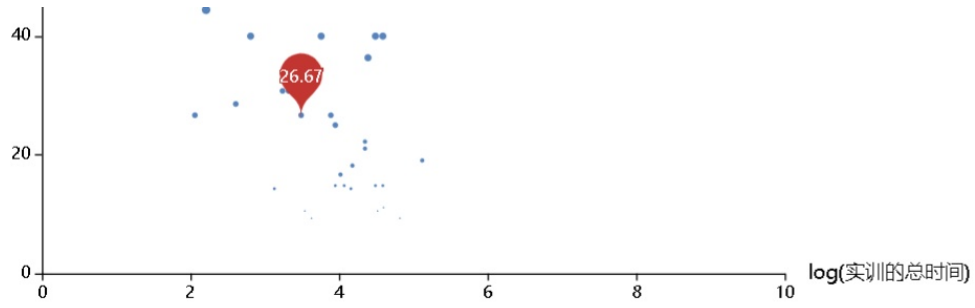
能力值

能力值（实训获得经验值/实训评测次数）

姓名	秦慧颖
学号	2340850057
能力	26.67



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 创建顺序结构的二叉树

调分：20

学员代码 ⓘ　　代码修改行数 53　　seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
 struct seqTree
 {
    int n;
    char *root;
 };

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
 int i = 0;
 while(i!=T->n){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	1350.64	2024-05-21 11:01	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	308.32	2024-05-21 11:03	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.01	332.5	2024-05-21 11:07	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0	332.5	2024-05-21 11:08	查看代码
5	共有2组测试集，其中有2组测试结果不匹配	0	345.68	2024-05-21 11:08	查看代码
6	共有2组测试集，其中有2组测试结果不匹配	0.02	332.5	2024-05-21 11:10	查看代码
7	共有2组测试集，其中有2组测试结果不匹配	0.02	332.5	2024-05-21 11:15	查看代码

8	共有2组测试集，其中有2组测试结果不匹配	0	307	2024-05-27 21:29	查看代码
9	评测通过	0.03	307	2024-05-27 21:30	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 二叉树基本操作

调分：20

学员代码 代码修改行数 89 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义 顺序 存储 结构
 */
 struct seqTree
 {
   int n;
   char *root;
 };

/*辅助 功能：逐个 输出 顺序 表的元素，元素 之间 以空格 为界*/
void printTree(struct seqTree *T)
{
 int i = 0;
 while(i!=T->n){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	332.5	2024-05-21 11:17	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	325.34	2024-05-27 21:27	查看代码
3	评测通过	0.01	101.84	2024-05-27 21:28	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 二叉树的层序遍历

调分: 20

学员代码 代码修改行数 92 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
 struct seqTree
 {
   int n;
   char *root;
 };

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
 int i = 0;
 while(i!=T->n){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	156.56	2024-05-27 21:28	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 二叉树的先序遍历

调分：18

学员代码 代码修改行数 92 seqtree.c

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
/*
定义顺序存储结构
 */
 struct seqTree
 {
   int n;
   char *root;
 };

/*辅助功能：逐个输出顺序表的元素，元素之间以空格为界*/
void printTree(struct seqTree *T)
{
 int i = 0;
 while(i!=T->n){
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有1组测试结果不匹配	0.01	101.84	2024-05-27 21:28	查看代码
2	评测通过	0	332.4	2024-05-27 21:29	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 10:58	页面停留时长	37分 55秒	10	- -	- -	99.0/100.0	优秀

阶段成绩

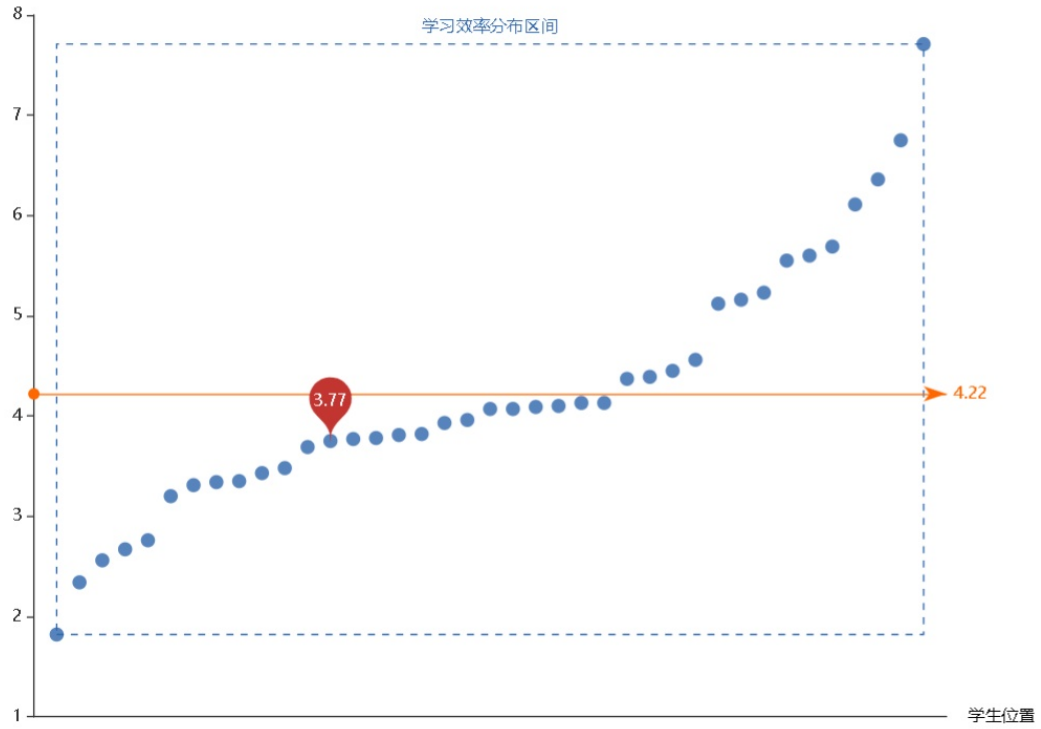
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-05-28 11:17	286	4	2024-06-04 10:50	21分 4秒	否	100/100	16.66/16.66	16.66
2	图的邻接表存储及图初始化	2024-05-28 11:38	104	1	2024-06-04 10:47	2分 6秒	否	300/300	16.66/16.66	16.66
3	图的广度遍历	2024-05-28 11:42	153	2	2024-06-04 10:45	3分 27秒	否	300/300	16.66/16.66	16.66
4	图的深度遍历	2024-05-28 11:43	194	1	2024-06-04 10:41	5分 8秒	否	300/300	16.66/16.66	16.66
5	拓扑排序	2024-06-04 10:41	253	1	2024-06-04 10:56	3分 36秒	否	300/300	16.66/16.66	16.66
6	prim算法生成最小生成树	2024-06-04	266	1	2024-06-04	2分 34秒	否	300/30	15.70/16.7	15.7

10:56 10:58 0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	李雯君
学号	2340850043
学习效率	3.77
当前排名	26

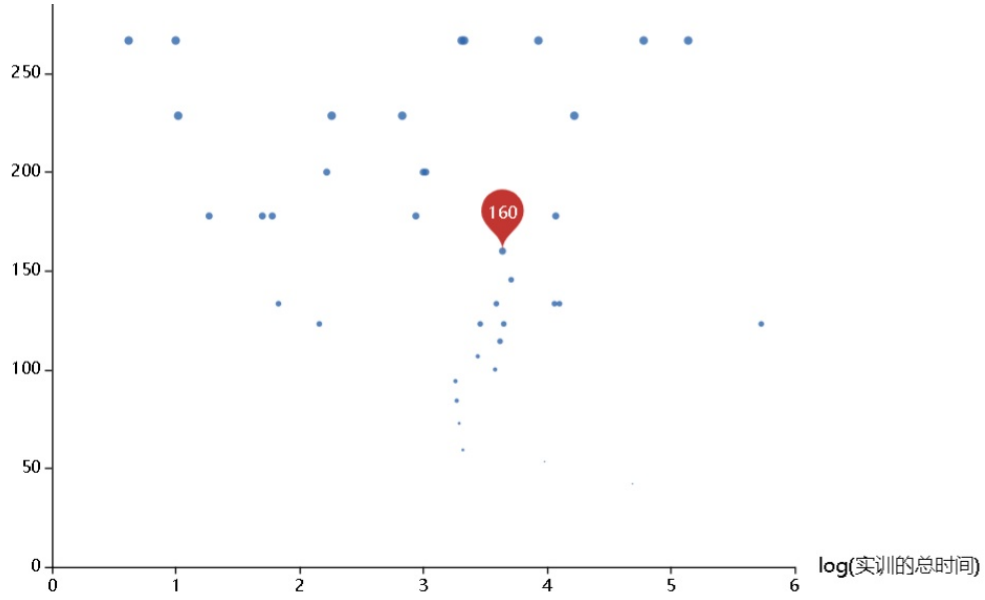
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	160

300



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：16.66

学员代码 ⓘ　　代码修改行数 286　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.03	116.08	2024-05-28 11:17	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2439.68	2024-05-28 11:38	查看代码
3	评测通过	0.01	50	2024-06-04 10:50	查看代码
4	评测通过	0.02	469.2	2024-06-04 10:51	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分：16.66

学员代码 ⓘ　　代码修改行数 104　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	234.6	2024-06-04 10:47	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 图的广度遍历

调分： 16.66

学员代码 ⓘ　　代码修改行数 153　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	190.76	2024-05-28 11:42	查看代码
2	评测通过	0.01	238.14	2024-06-04 10:45	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 图的深度遍历

调分：16.66

学员代码 ⓘ　　代码修改行数 194　　src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	225.51	2024-06-04 10:41	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 拓扑排序

调分: 16.66

学员代码 代码修改行数 253 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	190.31	2024-06-04 10:56	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分: 15.7

学员代码 代码修改行数 266 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	148.21	2024-06-04 10:58	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 11:27	页面停留时长	5时 8分 12秒	13	- -	- -	100.0/100.0	优秀

阶段成绩

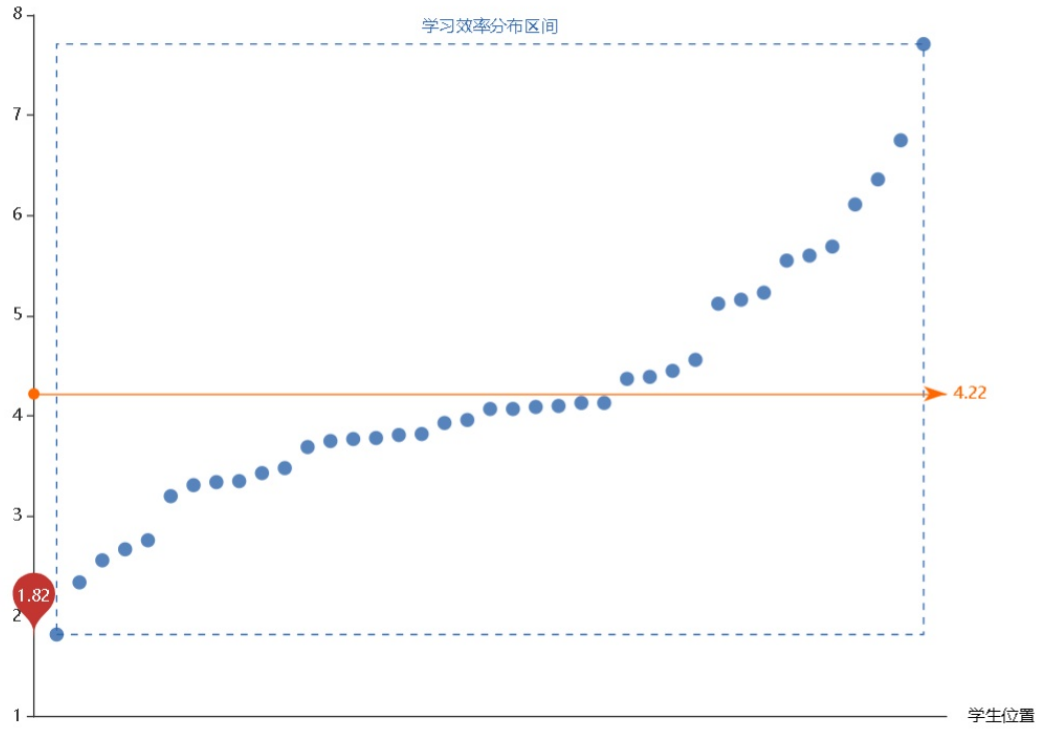
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-05-28 11:06	26	2	2024-05-28 11:24	18分 9秒	否	100/100	16.66/16.66	16.66
2	图的邻接表存储及图初始化	2024-05-28 11:24	86	1	2024-06-03 22:24	3时 7分 52秒	否	300/300	16.66/16.66	16.66
3	图的广度遍历	2024-05-29 15:01	153	4	2024-06-04 11:05	1时 20分 57秒	否	300/300	16.66/16.66	16.66
4	图的深度遍历	2024-06-04 11:05	195	3	2024-06-04 11:26	18分 6秒	否	300/300	16.66/16.66	16.66
5	拓扑排序	2024-06-04 11:20	253	1	2024-06-04 11:26	1分 30秒	否	300/300	16.66/16.66	16.66
6	prim算法生成最小生成树	2024-06-04	266	2	276 2024-06-04	1分 38秒	否	300/30	16.70/16.7	16.7

	11:23	11:27	0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	黄心媛
学号	2340850073
学习效率	1.82
当前排名	39

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

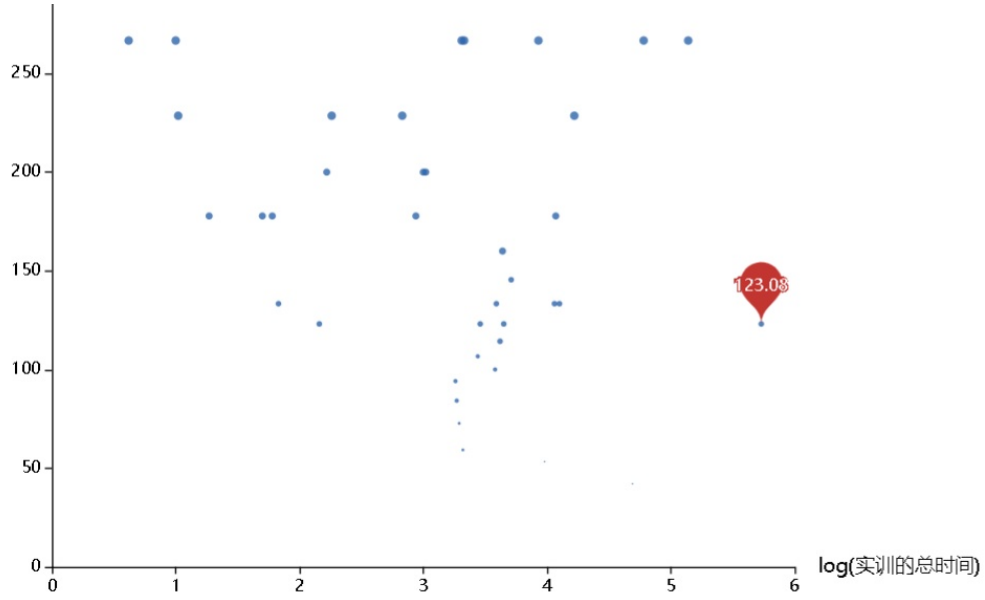
能力值

能力值（实训获得经验值/实训评测次数）

姓名	黄心媛
学号	2340850073
能力	123.08

277

300



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：16.66

学员代码 代码修改行数 26 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
//邻接矩阵数据结构
typedef  struct{
    int vcount;//顶点数
    int type ;//0 无向图，1 有向图
    char  vexs[N]  ;     // 顶点信息
    int  arcs[N][N]; //关系矩阵
} GraphMatrix;

//邻接表数据结构
struct EdgeNode { //边表中的结点
        int  endvex;     //相邻顶点在顶点表中下标
        int  weight;  //边的权
        struct EdgeNode  * nextedge;   //链字段
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	2439.6	2024-05-28 11:22	查看代码
2	评测通过	0.01	2439.84	2024-05-28 11:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分：16.66

学员代码 代码修改行数 86 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
//邻接矩阵数据结构
typedef  struct{
    int vcount;//顶点数
    int type ;//0 无向图，1 有向图
    char  vexs[N]  ;     // 顶点信息
    int  arcs[N][N]; //关系矩阵
} GraphMatrix;
//using namespace std;
//int visted[N] = { 0 };//存放访问过的顶点
//int v = 0;//默认从0开始遍历，可以修改
//邻接表数据结构
struct EdgeNode{ //边表中的结点
        int  endvex;     //相邻顶点在顶点表中下标
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	59.02	2024-06-03 22:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 图的广度遍历

调分：16.66

学员代码 ⓘ　　代码修改行数 153　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.02	1219.84	2024-06-04 10:28	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	171.68	2024-06-04 11:02	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0	159.05	2024-06-04 11:03	查看代码
4	评测通过	0.01	144.92	2024-06-04 11:05	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 图的深度遍历

调分：16.66

学员代码 ⓘ　　代码修改行数 195　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	192.93	2024-06-04 11:19	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	154.69	2024-06-04 11:25	查看代码
3	评测通过	0.01	589.3	2024-06-04 11:26	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 拓扑排序

调分：16.66

学员代码 代码修改行数 253 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	183.46	2024-06-04 11:26	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分：16.7

学员代码 代码修改行数 266 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	169.69	2024-06-04 11:27	查看代码
2	评测通过	0.01	60.29	2024-06-04 11:27	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



杨东泽 按时通关

学号：2340850041　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 11:46	页面停留时长	36分 4秒	12	- -	- -	89.0/100.0	良好

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-06-04 11:09	277	2	2024-06-04 11:40	31分 24秒	否	100/100	16.66/16.66	16.66
2	图的邻接表存储及图初始化	2024-06-04 11:40	104	3	2024-06-04 11:44	22秒	否	300/300	16.66/16.66	16.66
3	图的广度遍历	2024-06-04 11:40	153	3	2024-06-04 11:45	2分 56秒	否	300/300	15.66/16.66	15.66
4	图的深度遍历	2024-06-04 11:45	195	1	2024-06-04 11:45	26秒	否	300/300	12.66/16.66	12.66
5	拓扑排序	2024-06-04 11:45	253	2	2024-06-04 11:46	53秒	否	300/300	14.66/16.66	14.66
6	prim算法生成最小生成树	2024-06-04	266	1	2024-06-04	3秒	否	300/30	12.70/16.7	12.7

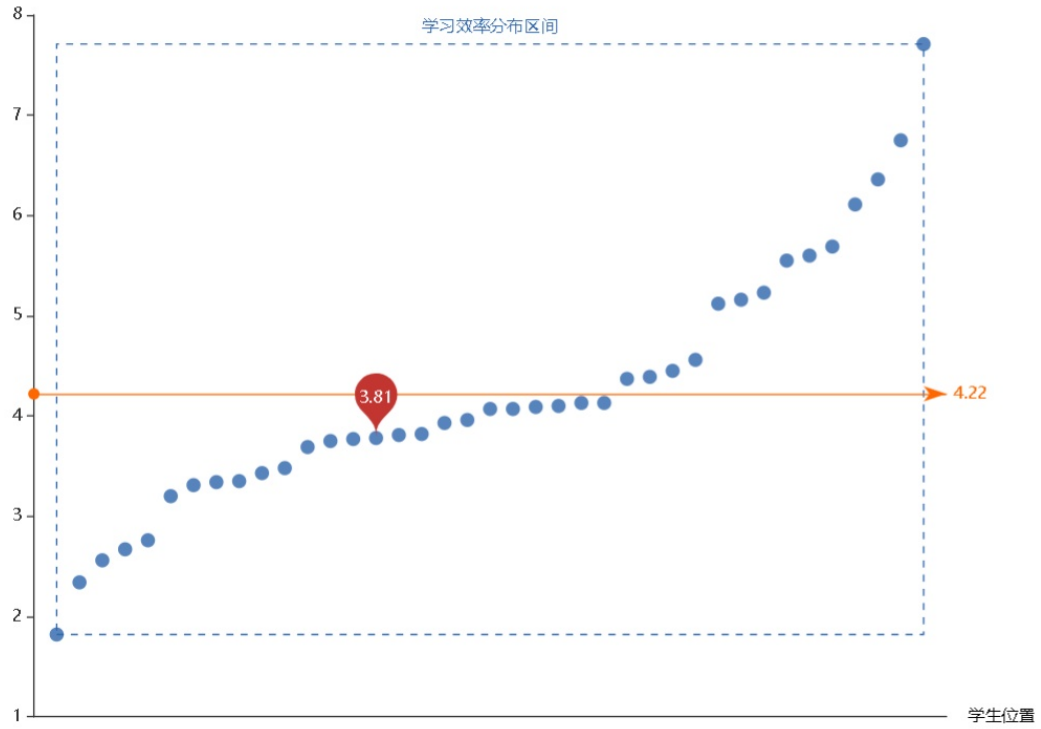
285

11:43 11:44 0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	杨东泽
学号	2340850041
学习效率	3.81
当前排名	24

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

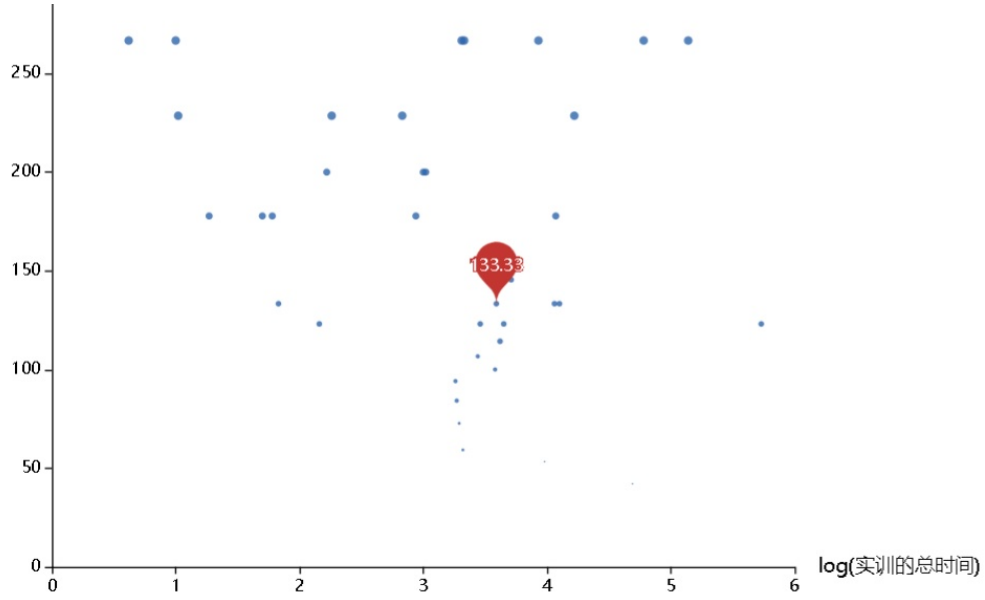
能力值

能力值（实训获得经验值/实训评测次数）

姓名	杨东泽
学号	2340850041
能力	133.33

286

300



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：16.66

学员代码 代码修改行数 277 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	2439.42	2024-06-04 11:34	查看代码
2	评测通过	0.02	1159.56	2024-06-04 11:40	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分：16.66

学员代码 ⓘ　　代码修改行数 104　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	162.91	2024-06-04 11:40	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	1219.78	2024-06-04 11:43	查看代码
3	评测通过	0.01	163.43	2024-06-04 11:44	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第3关 图的广度遍历

调分： 15.66

学员代码 代码修改行数 153 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	162.91	2024-06-04 11:40	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	1219.86	2024-06-04 11:43	查看代码
3	评测通过	0.01	579.78	2024-06-04 11:45	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 图的深度遍历

调分：12.66

学员代码 | 代码修改行数 195 | src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	1219.78	2024-06-04 11:45	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 拓扑排序

调分：14.66

学员代码 ⓘ　　代码修改行数 253　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	1220.1	2024-06-04 11:46	查看代码
2	评测通过	0.02	165.1	2024-06-04 11:46	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第6关 prim算法生成最小生成树

调分：12.7

学员代码 代码修改行数 266 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	1220.1	2024-06-04 11:44	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



于利友 按时通关

学号：2340850060　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-05 09:04	页面停留时长	50分 59秒	6	- -	- -	88.0/100.0	良好

阶段成绩

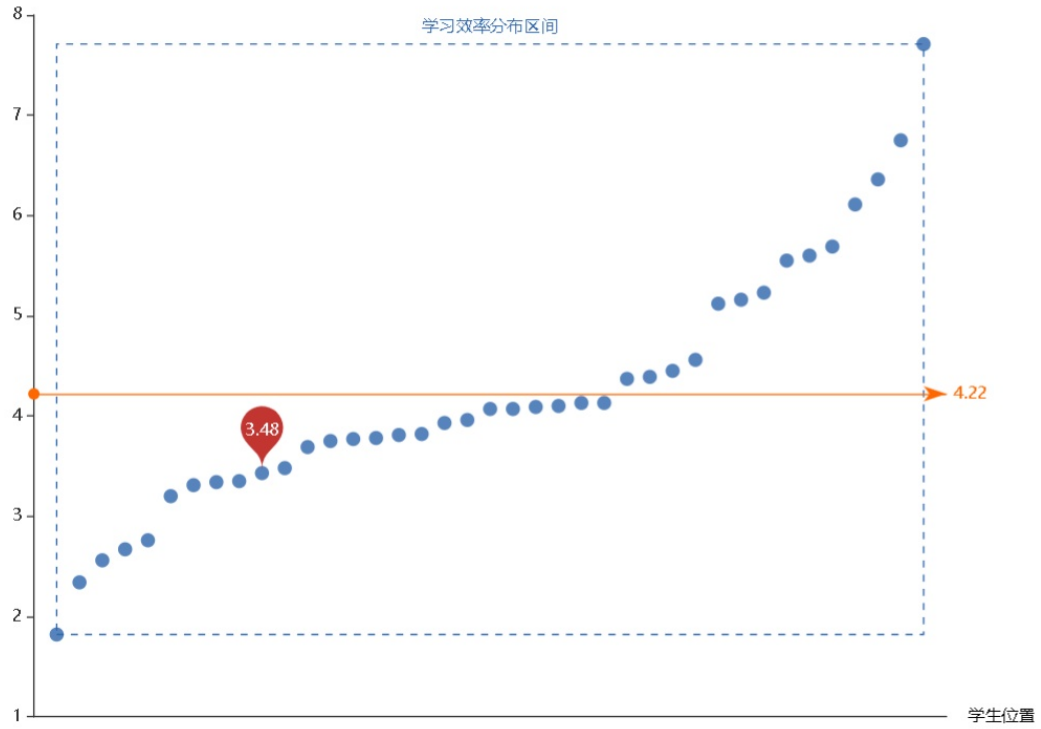
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-06-04 11:12	286	1	2024-06-05 08:18	5分 34秒	否	100/100	16.66/16.66	16.66
2	图的邻接表存储及图初始化	2024-06-05 08:18	104	1	2024-06-05 08:26	7分 44秒	否	300/300	16.66/16.66	16.66
3	图的广度遍历	2024-06-05 08:26	153	1	2024-06-05 08:36	10分 20秒	否	300/300	14.66/16.66	14.66
4	图的深度遍历	2024-06-05 08:36	195	1	2024-06-05 08:49	12分 17秒	否	300/300	12.66/16.66	12.66
5	拓扑排序	2024-06-05 08:49	253	1	2024-06-05 08:58	9分 31秒	否	300/300	12.66/16.66	12.66
6	prim算法生成最小生成树	2024-06-05 294	266	1	2024-06-05	5分 33秒	否	300/30	14.70/16.7	14.7

08:58 09:04 0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	于利友
学号	2340850060
学习效率	3.48
当前排名	29

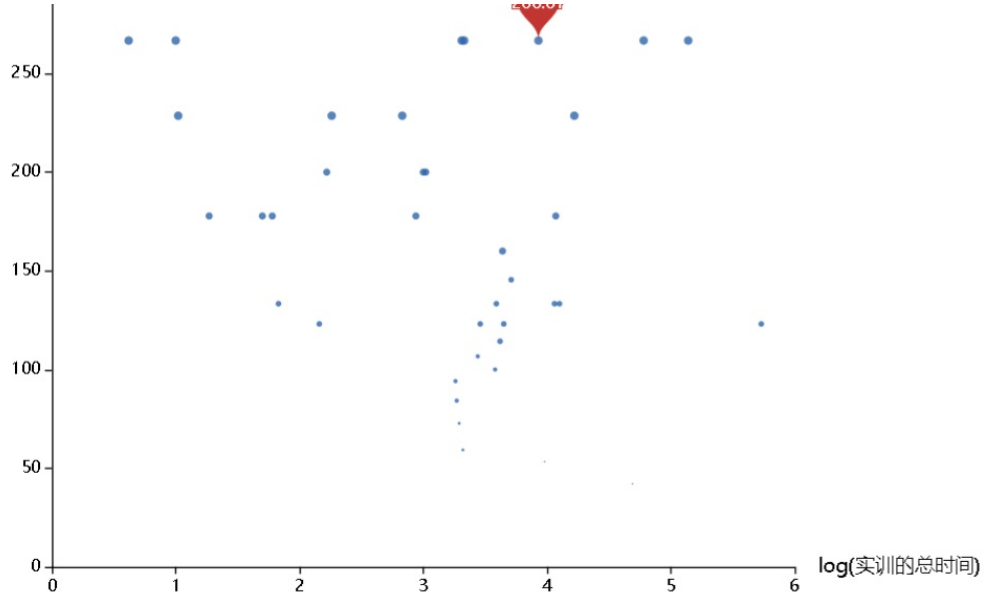
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）



姓名	于利友
学号	2340850060
能力	266.67



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分: 16.66

学员代码 代码修改行数 286 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	82.54	2024-06-05 08:18	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分: 16.66

学员代码 代码修改行数 104 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	101.52	2024-06-05 08:26	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 图的广度遍历

调分: 14.66

学员代码 代码修改行数 153 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	186.9	2024-06-05 08:36	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 图的深度遍历

调分: 12.66

学员代码 ⓘ　　代码修改行数 195　　src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	165.16	2024-06-05 08:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 拓扑排序

调分: 12.66

学员代码 | 代码修改行数 253 | src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	156.71	2024-06-05 08:58	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分: 14.7

学员代码 代码修改行数 266 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	153.52	2024-06-05 09:04	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



苏宁 补交通关

学号：2340850046　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-05 18:24	页面停留时长	2分 46秒	7	- -	- -	79.0/100.0	良好

阶段成绩

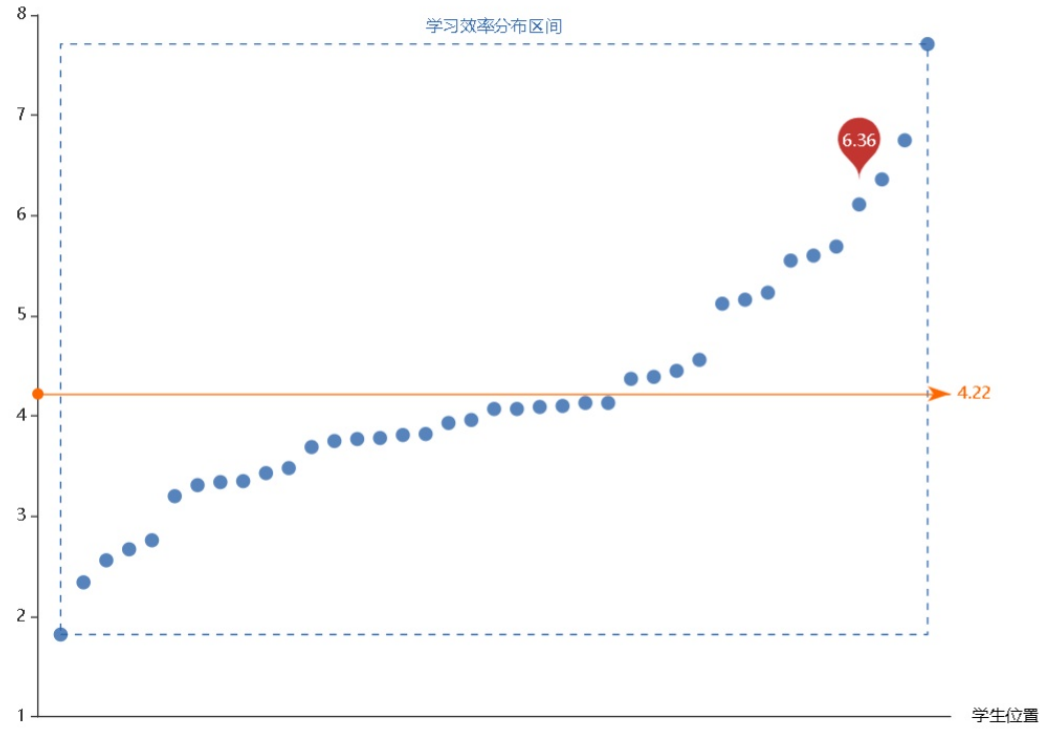
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-06-05 18:21	286	2	2024-06-05 18:22	1分 8秒	否	100/100	16.66/16.66	16.66
2	图的邻接表存储及图初始化	2024-06-05 18:22	104	1	2024-06-05 18:23	1分 4秒	否	300/300	12.66/16.66	12.66
3	图的广度遍历	2024-06-05 18:23	153	1	2024-06-05 18:23	9秒	否	300/300	16.66/16.66	16.66
4	图的深度遍历	2024-06-05 18:23	195	1	2024-06-05 18:23	10秒	否	300/300	10.66/16.66	10.66
5	拓扑排序	2024-06-05 18:23	253	1	2024-06-05 18:23	9秒	否	300/300	11.66/16.66	11.66
6	prim算法生成最小生成树	2024-06-05	266	1	2024-06-05	6秒	否	300/30	10.70/16.7	10.7

	18:23	18:24		0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	苏宁
学号	2340850046
学习效率	6.36
当前排名	3

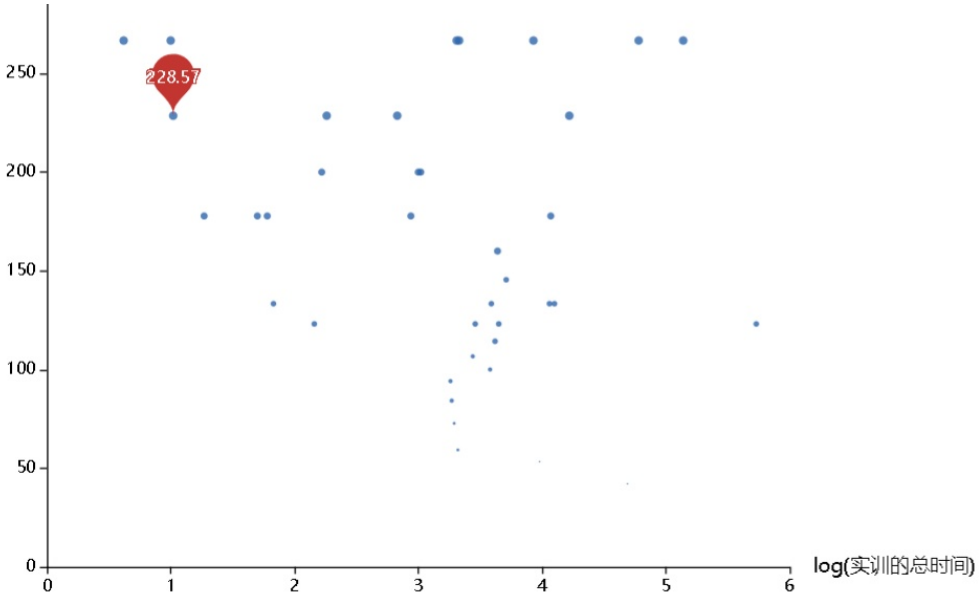
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名	苏宁
学号	2340850046
能力	228.57

300



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分： 16.66

学员代码 代码修改行数 286 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	309.28	2024-06-05 18:22	查看代码
2	评测通过 截止后评测	0.01	315.88	2024-06-05 18:22	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分: 12.66

学员代码 代码修改行数 104 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	156.96	2024-06-05 18:23	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 图的广度遍历

调分：16.66

学员代码 ⓘ　　代码修改行数 153　　src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	157.94	2024-06-05 18:23	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第4关 图的深度遍历

调分: 10.66

学员代码 ⓘ 代码修改行数 195 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	159.32	2024-06-05 18:23	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 拓扑排序

调分: 11.66

学员代码 代码修改行数 253 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	159.32	2024-06-05 18:23	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分：10.7

学员代码 代码修改行数 266 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过 截止后评测	0.01	159.32	2024-06-05 18:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



周雯浩 补交通关

学号：2340850047　　分班：未分班

截止前完成关卡：6/6　　最新完成关卡：6/6　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-12 10:36	页面停留时长	26分 52秒	22	- -	- -	78.0/100.0	良好

阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-06-05 09:14	46	7	2024-06-05 09:29	4分 44秒	否	100/100	10.66/16.66	10.66
2	图的邻接表存储及图初始化	2024-06-05 09:16	104	6	2024-06-12 10:35	7分 7秒	否	300/300	16.66/16.66	16.66
3	图的广度遍历	2024-06-05 09:30	153	2	2024-06-12 10:36	59秒	否	300/300	12.66/16.66	12.66
4	图的深度遍历	2024-06-05 09:30	195	2	2024-06-12 10:36	55秒	否	300/300	16.66/16.66	16.66
5	拓扑排序	2024-06-05 09:31	253	4	2024-06-12 10:36	12分 58秒	否	300/300	10.66/16.66	10.66
6	prim算法生成最小生成树	2024-06-05	285	1	2024-06-05	9秒	否	300/300	10.70/16.7	10.7

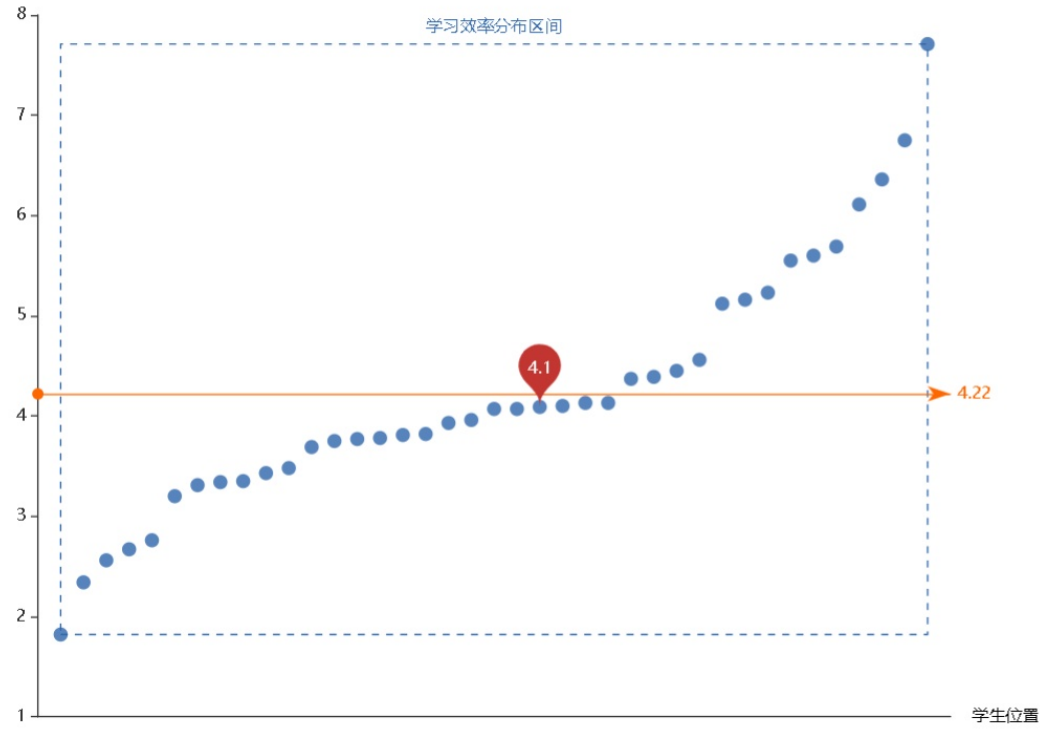
312

	09:18	09:31		0

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	周雯浩
学号	2340850047
学习效率	4.1
当前排名	17

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

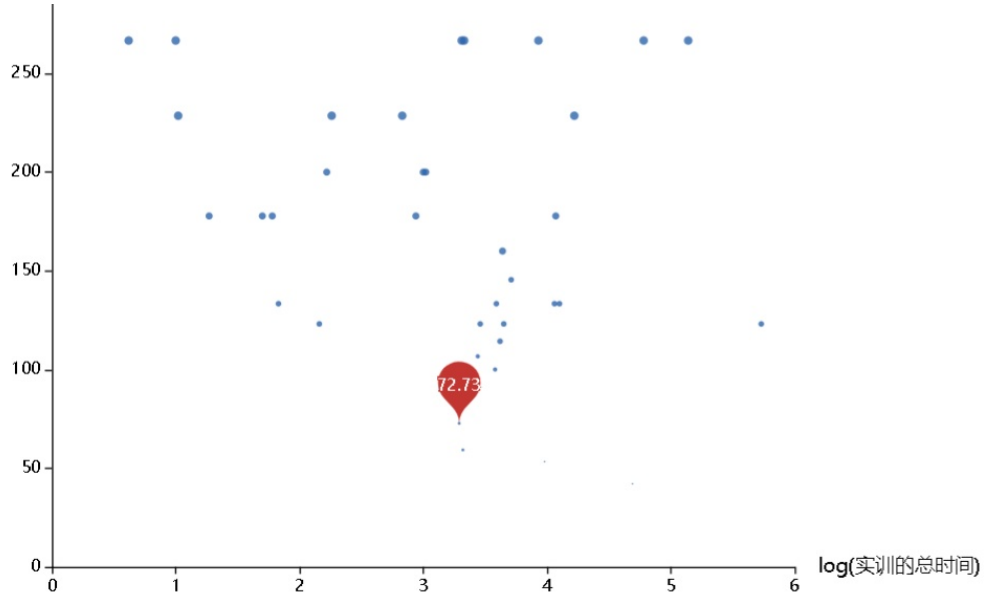
能力值

能力值（实训获得经验值/实训评测次数）

姓名	周雯浩
学号	2340850047
能力	72.73

313

300



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：10.66

学员代码 代码修改行数 46 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	318.62	2024-06-05 09:15	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2439.72	2024-06-05 09:17	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	2440.34	2024-06-05 09:19	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0	2439.72	2024-06-05 09:22	查看代码
5	评测通过	0.01	319.74	2024-06-05 09:29	查看代码
6	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	350.9	2024-06-11 11:33	查看代码
7	共有2组测试集，其中有2组测试结果不匹配 截止后评测	0	350.9	2024-06-11 11:36	查看代码

第2关 图的邻接表存储及图初始化

调分：16.66

学员代码 ⓘ　　代码修改行数 104　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	161.65	2024-06-05 09:18	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	1219.82	2024-06-05 09:20	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0.01	161.77	2024-06-05 09:21	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0	1220.04	2024-06-05 09:21	查看代码
5	共有1组测试集，其中有1组测试结果不匹配	0.03	1220.04	2024-06-05 09:29	查看代码
6	评测通过 截止后评测	0.01	435.68	2024-06-12 10:35	查看代码

代码评价　　一键AI生成　编辑

第3关 图的广度遍历

调分： 12.66

学员代码 代码修改行数 153 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.04	163.93	2024-06-05 09:30	查看代码
2	评测通过 截止后评测	0.01	435.68	2024-06-12 10:36	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 图的深度遍历

调分：16.66

学员代码 ⓘ　　代码修改行数 195　　src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	0	2024-06-05 09:31	查看代码
2	评测通过 截止后评测	0.01	435.68	2024-06-12 10:36	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第5关 拓扑排序

调分： 10.66

学员代码 代码修改行数 253 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	44.56	2024-06-05 09:38	查看代码
2	共有1组测试集，其中有1组测试结果不匹配 截止后评测	0	194.22	2024-06-11 11:29	查看代码
3	共有1组测试集，其中有1组测试结果不匹配 截止后评测	0	218.84	2024-06-11 11:30	查看代码
4	评测通过 截止后评测	0.01	223.2	2024-06-12 10:36	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分：10.7

学员代码 代码修改行数 285 src/graph.h

复制代码

```c
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	818.19	2024-06-05 09:31	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



陈家明 未通关

学号：2340850010　　分班：未分班

截止前完成关卡：1/6　　最新完成关卡：1/6　　完成效率：0.00　　课堂最高完成效率：10.12

通关时间	计时规则	实训总耗时	评测次数	关卡得分	效率分	查重扣分	补交扣分	最终成绩	总评
--	页面停留时长	20分 36秒	9	13.3/80.0	0.0/20.0	- -	- -	13.3/100.0	不及格

阶段成绩

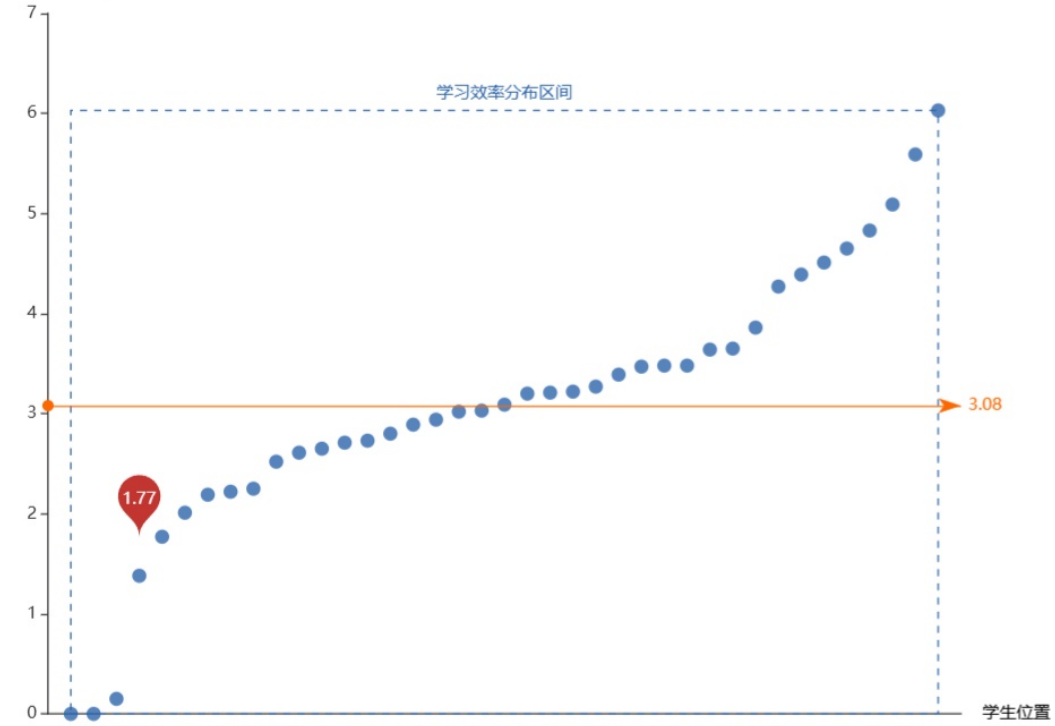
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-05-16 13:26	275	4	2024-05-16 13:47	20分 36秒	否	100/100	13.30/13.3	13.3
2	图的邻接表存储及图初始化	2024-05-16 13:47	- -	2	--	--	否	0/300	0.00/13.3	0
3	图的广度遍历	2024-05-16 13:49	- -	1	--	--	否	0/300	0.00/13.3	0
4	图的深度遍历	2024-05-16 13:49	- -	2	--	--	否	0/300	0.00/13.3	0
5	拓扑排序	--	- -	0	--	--	否	0/300	0.00/13.3	0
6	prim算法生成最小生成树	--	- -	0	--	--	否	0/300	0.00/13.5	0

321

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	陈家明
学号	2340850010
学习效率	1.77
当前排名	35

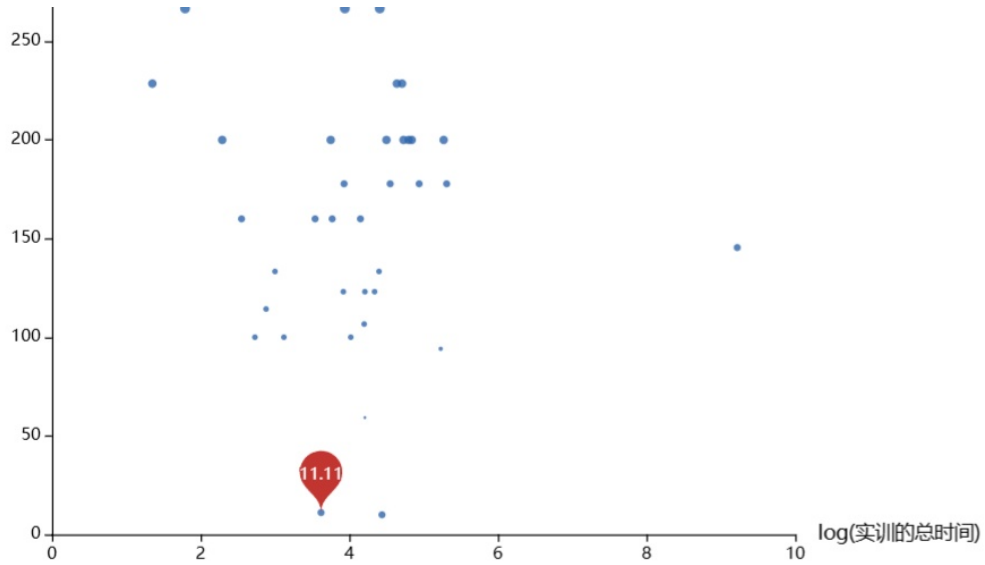
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）

姓名	陈家明
学号	2340850010
能力	11.11





1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：13.3

学员代码 ⓘ　　代码修改行数 275　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
    QueueNode *tail;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	324.28	2024-05-16 13:36	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	324.28	2024-05-16 13:43	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	2072.4	2024-05-16 13:46	查看代码
4	评测通过	0.03	109.24	2024-05-16 13:47	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分: 0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	1044.5	2024-05-16 13:47	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0.01	1036.2	2024-05-16 13:48	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 图的广度遍历

调分: 0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0.01	1044.5	2024-05-16 13:49	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 图的深度遍历

调分: 0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	0	2024-05-16 13:49	查看代码
2	共有1组测试集，其中有1组测试结果不匹配 截止后评测	0	0	2024-06-21 16:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 拓扑排序

调分: 0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

本关无评测记录

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分: 0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

本关无评测记录

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



宋政廷 未通关

学号：2340850028　　分班：未分班

截止前完成关卡：1/6　　最新完成关卡：1/6　　完成效率：0.00　　课堂最高完成效率：10.12

通关时间	计时规则	实训总耗时	评测次数	关卡得分	效率分	查重扣分	补交扣分	最终成绩	总评
--	页面停留时长	33分 41秒	10	13.3/80.0	0.0/20.0	- -	- -	13.3/100.0	不及格

阶段成绩

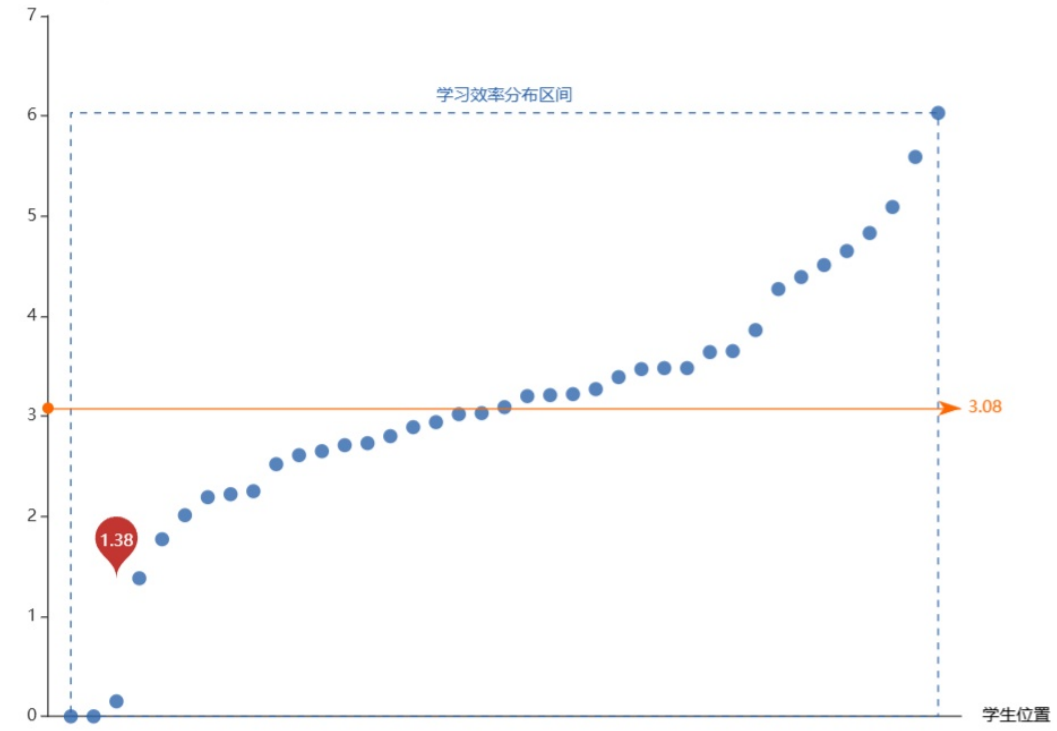
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	图的邻接矩阵存储及图初始化	2024-05-09 13:39	27	1	2024-05-09 14:13	33分 41秒	否	100/100	13.30/13.3	13.3
2	图的邻接表存储及图初始化	2024-05-09 14:13	- -	6	--	--	否	0/300	0.00/13.3	0
3	图的广度遍历	2024-05-09 14:16	- -	2	--	--	否	0/300	0.00/13.3	0
4	图的深度遍历	2024-05-09 15:04	- -	1	--	--	否	0/300	0.00/13.3	0
5	拓扑排序	2024-05-09 15:05	- -	0	--	--	否	0/300	0.00/13.3	0
6	prim算法生成最小生成树	--	- -	0	--	--	否	0/300	0.00/13.5	0

330

图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	宋政廷
学号	2340850028
学习效率	1.38
当前排名	36

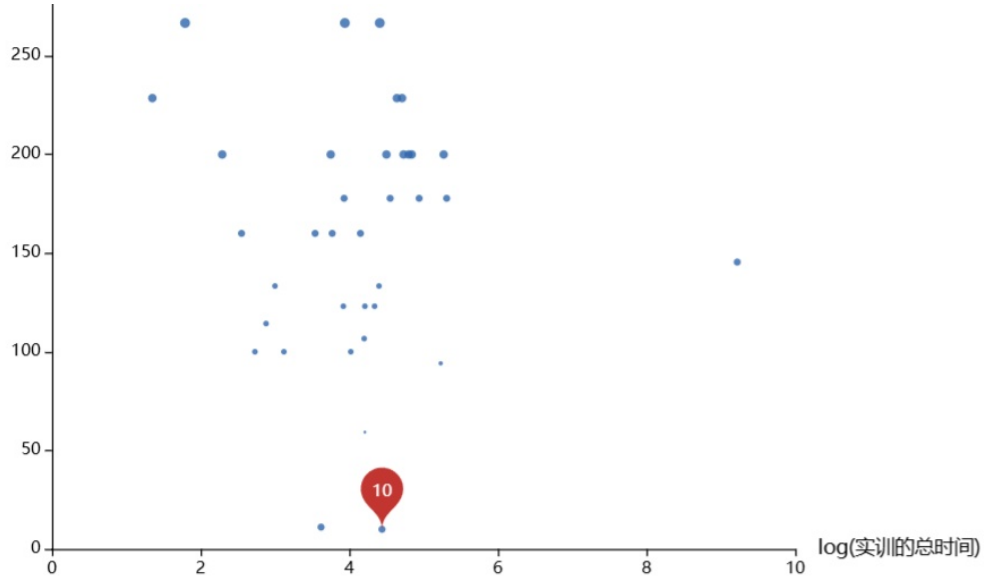
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

能力值

能力值（实训获得经验值/实训评测次数）



姓名	宋政廷
学号	2340850028
能力	10



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少

实训详情

第1关 图的邻接矩阵存储及图初始化

调分：13.3

学员代码 ⓘ　　代码修改行数 27　　src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
//邻接矩阵数据结构
typedef  struct{
   int vcount;//顶点数
   int type ;//0 无向图，1 有向图
   char  vexs[N]  ;     // 顶点信息
   int  arcs[N][N]; //关系矩阵
} GraphMatrix;

//邻接表数据结构
struct EdgeNode { //边表中的结点
      int  endvex;     //相邻顶点在顶点表中下标
      int  weight;  //边的权
      struct EdgeNode  * nextedge;   //链字段
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	1185.46	2024-05-09 14:13	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第2关 图的邻接表存储及图初始化

调分：0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	988.08	2024-05-09 14:14	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	988.08	2024-05-09 14:15	查看代码
3	共有1组测试集，其中有1组测试结果不匹配	0	461.29	2024-05-09 14:18	查看代码
4	共有1组测试集，其中有1组测试结果不匹配	0	323.7	2024-05-09 14:18	查看代码
5	共有1组测试集，其中有1组测试结果不匹配	0.01	53.51	2024-05-09 15:04	查看代码
6	共有1组测试集，其中有1组测试结果不匹配	0	71.04	2024-05-09 15:05	查看代码

代码评价 一键AI生成 编辑

第3关 图的广度遍历

调分：0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	1219.88	2024-05-09 14:16	查看代码
2	共有1组测试集，其中有1组测试结果不匹配	0	49.65	2024-05-09 15:04	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第4关 图的深度遍历

调分：0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有1组测试集，其中有1组测试结果不匹配	0	0	2024-05-09 15:05	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第5关 拓扑排序

调分：0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

本关无评测记录

代码评价

一键AI生成 编辑

暂无评价~

第6关 prim算法生成最小生成树

调分：0

学员代码 src/graph.h

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#define N  6
#define MAX 100
typedef struct {
    int vex;
    int shortweight;
}ShortWeight;
typedef struct QueueNode{
    int val;
    struct QueueNode* next;
}QueueNode;

typedef struct Queue{
    QueueNode *head;
```

本关无评测记录

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 11:26	页面停留时长	29分 30秒	7	- -	- -	95.0/100.0	优秀

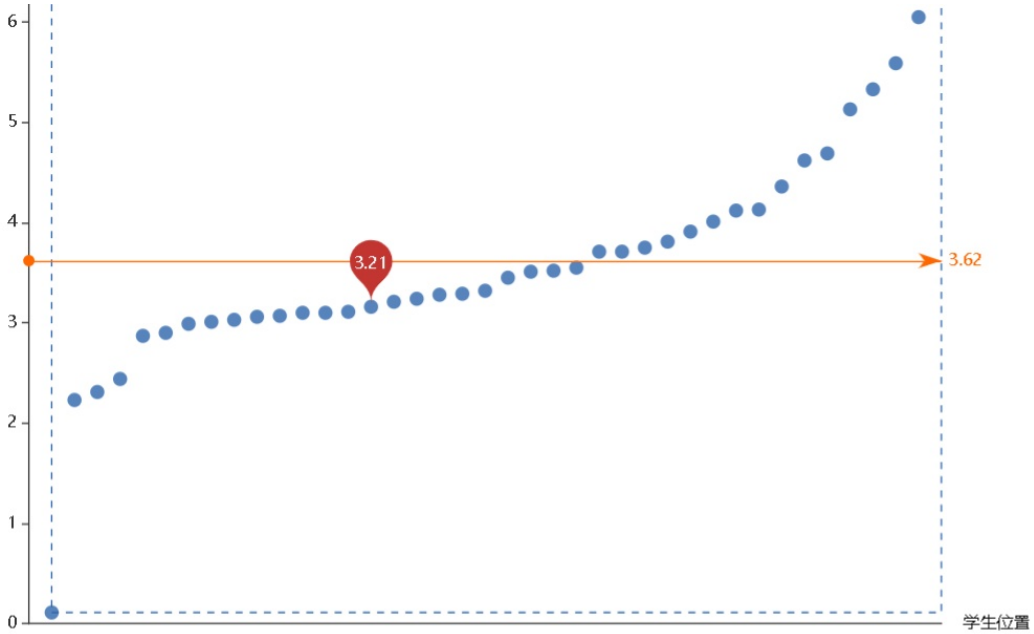
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-21 11:19	8	4	2024-06-04 11:06	13分 35秒	否	100/100	33.33/33.33	33.33
2	查找-进阶(二分查找)	2024-06-04 10:53	17	2	2024-06-04 11:25	15分 21秒	否	300/300	31.33/33.33	31.33
3	查找进阶	2024-06-04 10:53	13	1	2024-06-04 11:26	34秒	否	300/300	30.34/33.34	30.34

图形统计



姓名　李雯君
学号　2340850043
学习效率　3.21
当前排名　25

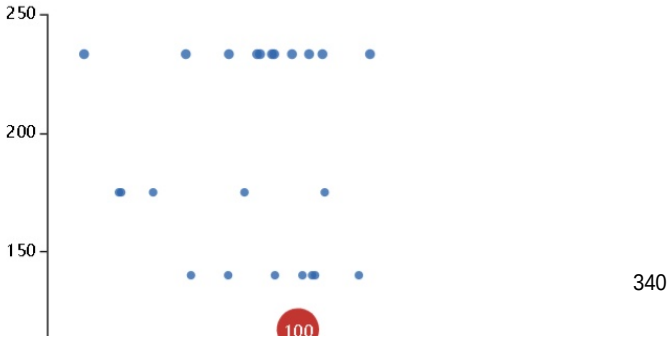


1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

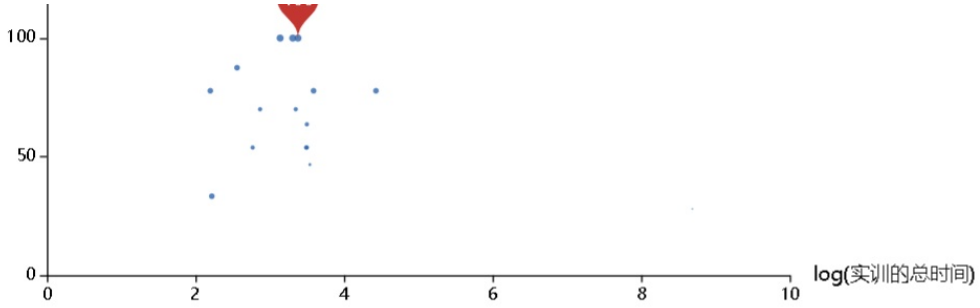
能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	100



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 33.33

学员代码 代码修改行数 8 src/step1/find_pos.c

复制代码

```
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
  int i;
  for(i=0;i<n;i++)
  {
    if(data[i]==key)
      return i;
  }
  return -1;
 /*------------end-----------------*/
}

int main(void)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有4组测试集，其中有2组测试结果不匹配	0.01	767.12	2024-06-04 11:04	查看代码
2	共有4组测试集，其中有2组测试结果不匹配	0.03	616.44	2024-06-04 11:05	查看代码
3	共有4组测试集，其中有2组测试结果不匹配	0.04	626.84	2024-06-04 11:05	查看代码
4	评测通过	0.01	621.24	2024-06-04 11:06	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 查找-进阶(二分查找)

调分: 31.33

学员代码 代码修改行数 17 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数.

/*------------begin-----------------*/
int l,r,mid;
int count = 0;
l = 0,r = n-1;
while(l<=r)
{
 mid = (l+r)/2;
 if(key<data[mid])r = mid-1;
 else if(key>data[mid]) l = mid+1;
 else return ++count;
 count++;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有5组测试集，其中有5组测试结果不匹配	0	0	2024-06-04 11:18	查看代码
2	评测通过	0.07	819.2	2024-06-04 11:25	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分：30.34

学员代码 ⓘ　　代码修改行数 13　　src/step3/array_val.c

复制代码

```
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin----------*/
  for(int i = 0;i<n;i++){
    for(int j = i+1;j<n;j++){
      if(data[i]+data[j]==target)
      {
        printf("%d,%d",i,j);
        return;
      }
    }
  }
  printf("-1,-1");
  return;
  /*----------end----------*/
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	624.88	2024-06-04 11:26	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-28 11:05	页面停留时长	35分 34秒	5	- -	- -	96.0/100.0	优秀

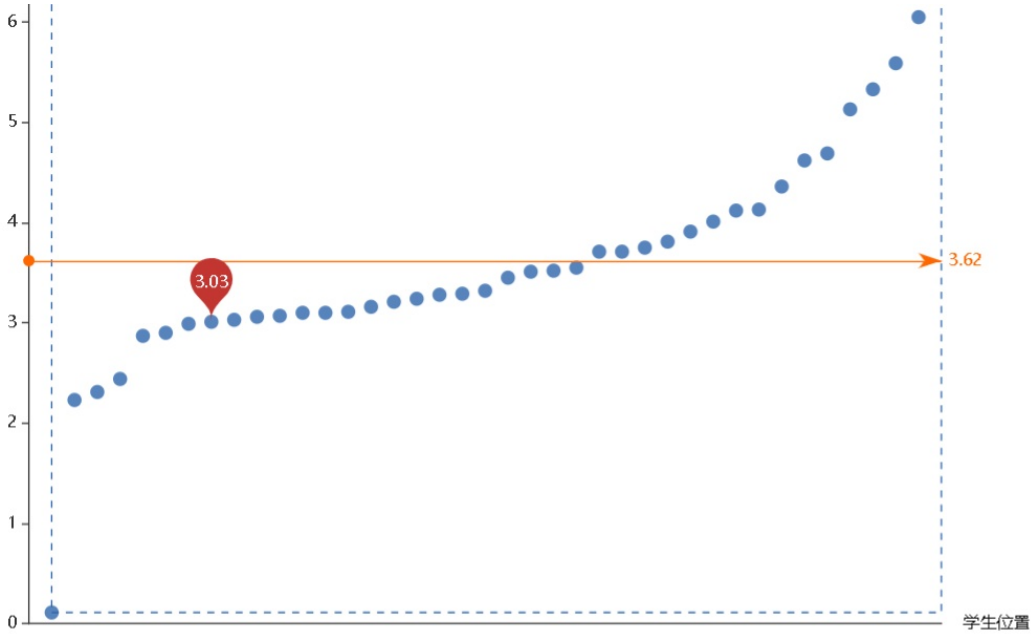
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-23 21:13	9	1	2024-05-28 10:36	6分 24秒	否	100/100	33.33/33.33	33.33
2	查找-进阶(二分查找)	2024-05-28 10:36	18	3	2024-05-28 10:56	20分 15秒	否	300/300	31.33/33.33	31.33
3	查找进阶	2024-05-28 10:56	12	1	2024-05-28 11:05	8分 55秒	否	300/300	31.34/33.34	31.34

图形统计



姓名　黄心媛
学号　2340850073
学习效率　3.03
当前排名　32

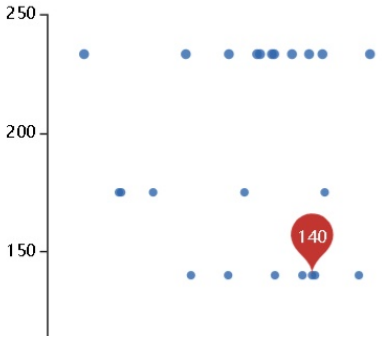


1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

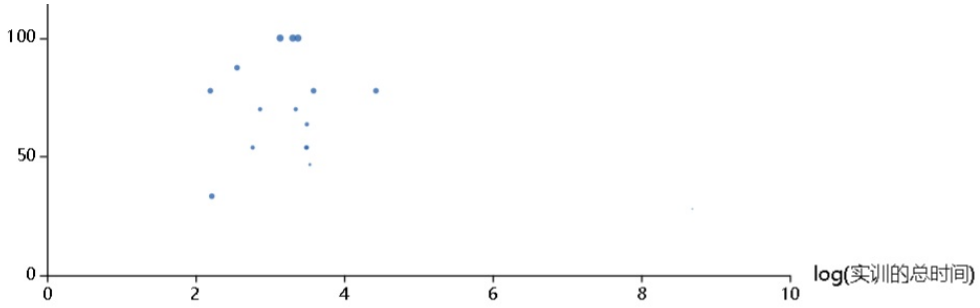
能力值

能力值（实训获得经验值/实训评测次数）

姓名	黄心媛
学号	2340850073
能力	140



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 33.33

学员代码 代码修改行数 9 src/step1/find_pos.c

复制代码

```
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
 for(int i=0;i<n;i++)
 {
   if(data[i]==key)
     return i;
 }
 return -1;

 /*------------end-----------------*/
}

int main(void)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0	4880.12	2024-05-28 10:36	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 查找-进阶(二分查找)

调分: 31.33

学员代码 代码修改行数 18 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数.

/*------------begin-----------------*/
int left=0,right=n-1;
int middle;
int count=0;
while(left<=right)
{
    count++;
    middle=(left+right)/2;
    if(data[middle]>key)
        right=middle-1;
    else if(data[middle]<key)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有5组测试集，其中有5组测试结果不匹配	0	0	2024-05-28 10:43	查看代码
2	共有5组测试集，其中有2组测试结果不匹配	0	6099.2	2024-05-28 10:50	查看代码
3	评测通过	0.01	1283.3	2024-05-28 10:56	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分：31.34

学员代码 ⓘ　　代码修改行数 12　　src/step3/array_val.c

复制代码

```c
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin----------*/

  for(int i=0;i<n;i++)
  {
    for(int j=i+1;j<n;j++)
    {
      if(data[i]+data[j]==target)
      {
        printf("%d,%d",i,j);
        return 0;
      }
    }
  }
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	4879.68	2024-05-28 11:05	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



周钰欣 按时通关

学号：2340850045　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-28 10:25	页面停留时长	31分 3秒	5	- -	- -	88.0/100.0	良好

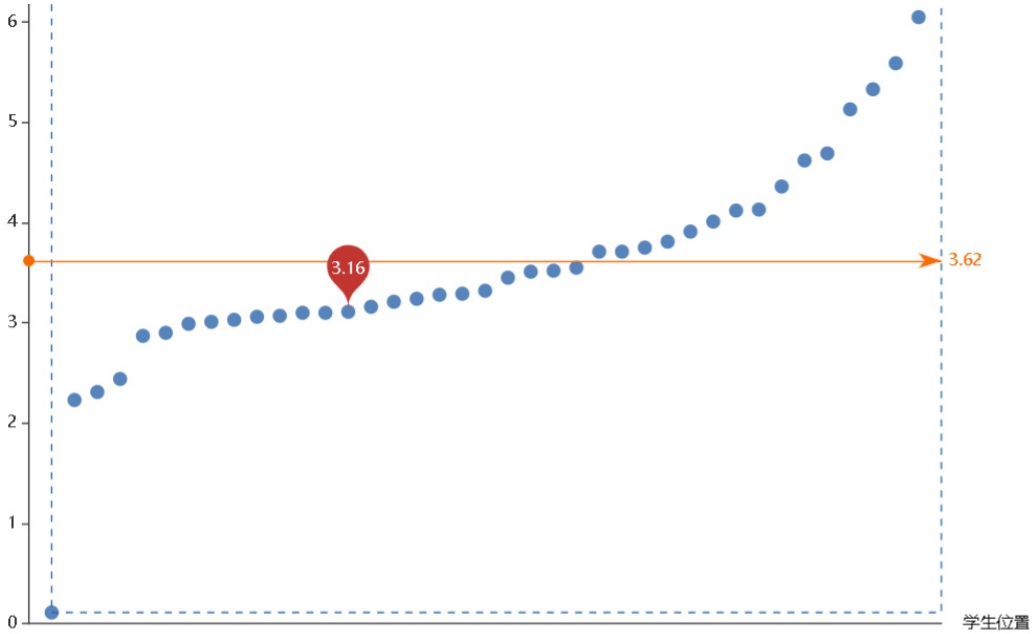
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-21 10:44	9	1	2024-05-21 10:48	4分 7秒	否	100/100	31.33/33.33	31.33
2	查找-进阶(二分查找)	2024-05-21 10:49	22	2	2024-05-21 10:52	3分 32秒	否	300/300	28.33/33.33	28.33
3	查找进阶	2024-05-21 11:24	9	2	2024-05-28 10:25	23分 24秒	否	300/300	28.34/33.34	28.34

图形统计



姓名	周钰欣
学号	2340850045
学习效率	3.16
当前排名	26

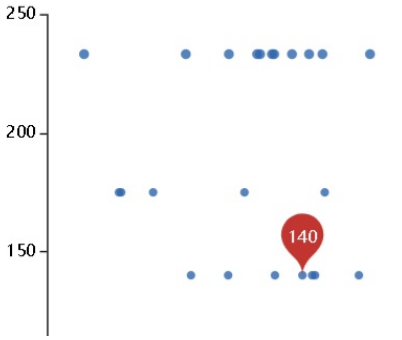


1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

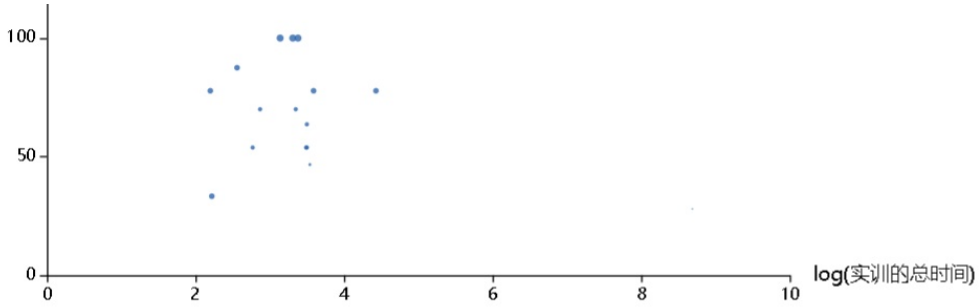
能力值

能力值（实训获得经验值/实训评测次数）

姓名	周钰欣
学号	2340850045
能力	140



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 31.33

学员代码 代码修改行数 9 src/step1/find_pos.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
  for(int i=0;i<n;i++){
    if(data[i]==key){
      return i;
    }
  }
  return -1;
 /*------------end-----------------*/
}

int main(void)
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	658.32	2024-05-21 10:48	查看代码

代码评价

一键AI生成 编辑

暂无评价~

调分记录

第2关 查找-进阶(二分查找)

调分: 28.33

学员代码 代码修改行数 22 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数.

 /*------------begin-----------------*/
  int low=0,high=n-1,count=0;
  if(key<data[0]||key>data[n-1]){
    return count;
  }
  while(low<=high){
    int mid=(low+high)/2;
    if(key==data[mid]){
      count++;
      return count;
    }
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	826.1	2024-05-21 10:52	查看代码
2	评测通过	0.02	811.8	2024-05-21 11:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分: 28.34

学员代码 ⓘ　　代码修改行数 9　　src/step3/array_val.c

复制代码

```
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin--------------*/
  int a=-1,b=-1;
  for(int i=0;i<n-1;i++){
    for(int j=i+1;j<n;j++){
      if(data[i]+data[j]==target){
        a=i;b=j;
      }
    }
  }
  printf("%d,%d",a,b);
 /*----------end--------------*/

}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有4组测试集，其中有4组测试结果不匹配	0	703.8	2024-05-28 10:24	查看代码
2	评测通过	0	1122.28	2024-05-28 10:25	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



刘宣宣 按时通关

学号：2340850065　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 00:30	页面停留时长	32分 40秒	13	- -	- -	86.0/100.0	良好

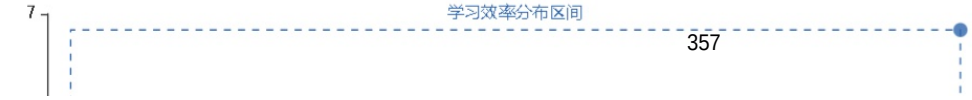
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-26 23:07	7	7	2024-06-04 00:27	27分 14秒	否	100/100	29.33/33.33	29.33
2	查找-进阶(二分查找)	2024-06-03 09:21	15	2	2024-06-04 00:29	1分 47秒	否	300/300	28.33/33.33	28.33
3	查找进阶	2024-06-03 09:21	14	4	2024-06-04 00:30	3分 39秒	否	300/300	28.34/33.34	28.34

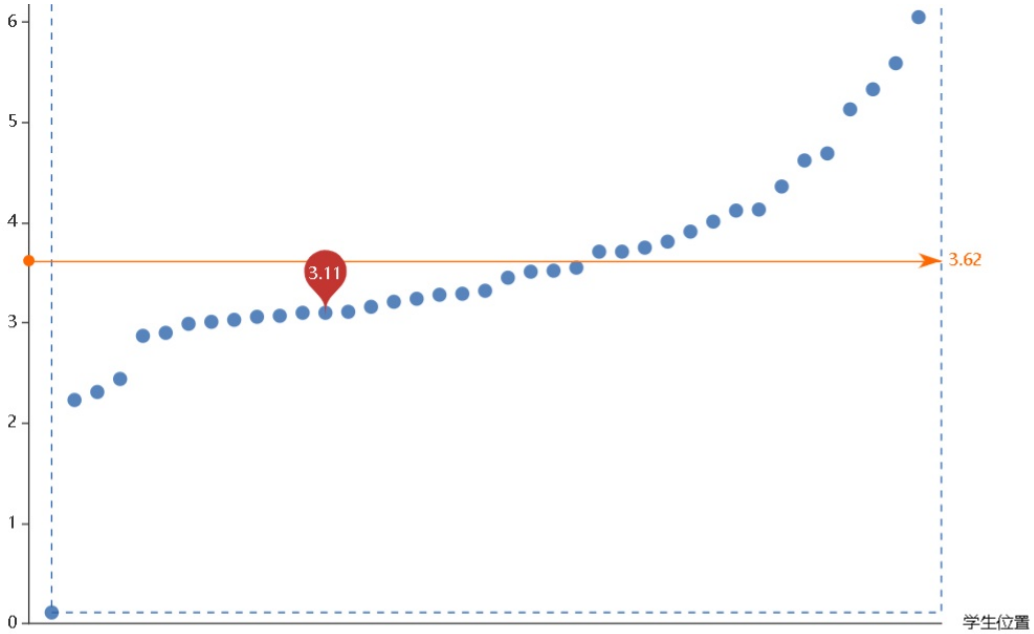
图形统计

学习效率

学习效率=log(实训总得分/实训总耗时)



姓名	刘宣宣
学号	2340850065
学习效率	3.11
当前排名	27

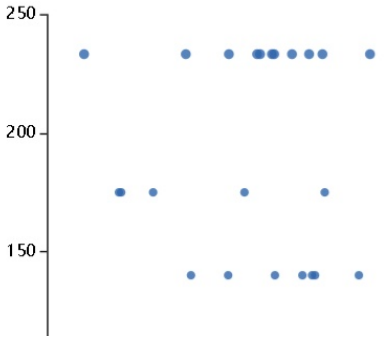


1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

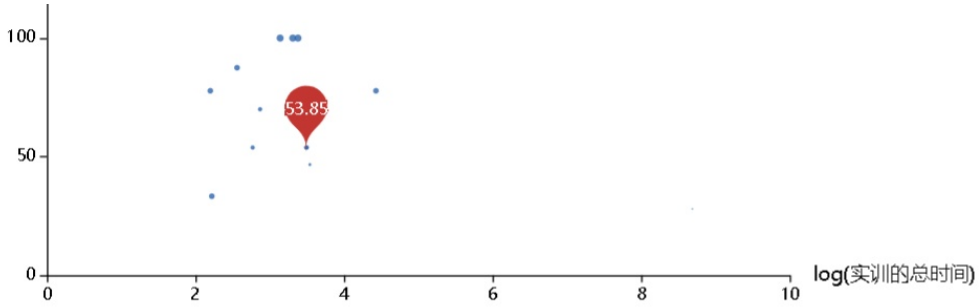
能力值

能力值（实训获得经验值/实训评测次数）

姓名	刘宣宣
学号	2340850065
能力	53.85



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 29.33

学员代码 代码修改行数 7 src/step1/find_pos.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
  for(int i = 0;i<n;i++){
    if(data[i]==key)return i;
  }
  return -1;
 /*------------end-----------------*/
}

int main(void)
{
int a[]={99,23,5,189,-1,66};
int x;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有4组测试集，其中有3组测试结果不匹配	0.05	278.92	2024-05-26 23:09	查看代码
2	共有4组测试集，其中有4组测试结果不匹配	0	698.16	2024-06-03 09:22	查看代码
3	共有4组测试集，其中有4组测试结果不匹配	0	200.96	2024-06-04 00:01	查看代码
4	共有4组测试集，其中有4组测试结果不匹配	0	204.56	2024-06-04 00:04	查看代码
5	共有4组测试集，其中有4组测试结果不匹配	0	594.28	2024-06-04 00:05	查看代码
6	共有4组测试集，其中有4组测试结果不匹配	0	594.28	2024-06-04 00:06	查看代码
7	评测通过	0.01	168.44	2024-06-04 00:27	查看代码

第2关 查找-进阶(二分查找)

调分: 28.33

学员代码 代码修改行数 15 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数.

 /*------------begin----------------*/
 int l,r,mid;
 int count = 0;
 l = 0,r = n-1;
 while(l<=r){
    mid = (l+r)/2;
    if(key<data[mid])r = mid-1;
    else if(key>data[mid]) l = mid+1;
    else return ++count;
    count++;
 }
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有5组测试集，其中有5组测试结果不匹配	0.01	628.2	2024-06-03 09:24	查看代码
2	评测通过	0.04	231.9	2024-06-04 00:29	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分: 28.34

学员代码 | 代码修改行数 14 | src/step3/array_val.c

复制代码

```
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin----------*/
  for(int i = 0;i<n;i++){
    for(int j = i+1;j<n;j++){
      if(data[i]+data[j]==target)
      {
        printf("%d,%d",i,j);
        return;
      }
    }
  }
  printf("-1,-1");
  return;
 /*----------end----------*/
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有4组测试集，其中有4组测试结果不匹配	0	502.56	2024-06-03 09:21	查看代码
2	共有4组测试集，其中有4组测试结果不匹配	0	207.24	2024-06-04 00:04	查看代码
3	评测通过	0.04	185.52	2024-06-04 00:30	查看代码
4	评测通过	0.06	641.8	2024-06-04 10:33	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



胡可 按时通关

学号：2340850042　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 11:49	页面停留时长	20分 37秒	3	- -	- -	77.0/100.0	良好

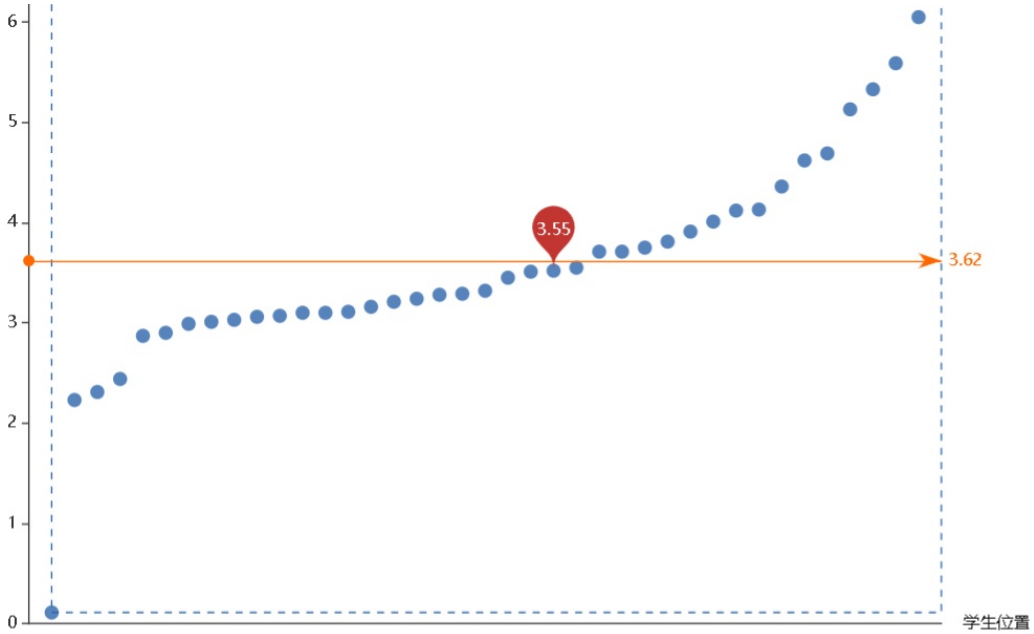
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-21 10:51	7	1	2024-05-21 11:48	19分 49秒	否	100/100	25.33/33.33	25.33
2	查找-进阶(二分查找)	2024-05-21 11:48	15	1	2024-05-21 11:48	29秒	否	300/300	25.33/33.33	25.33
3	查找进阶	2024-05-21 11:48	13	1	2024-05-21 11:49	19秒	否	300/300	26.34/33.34	26.34

图形统计



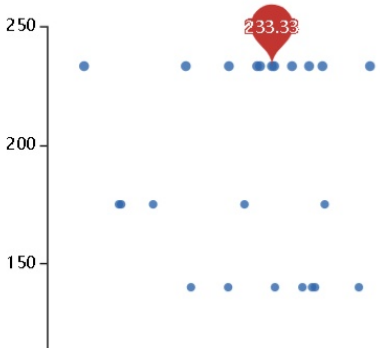
姓名	胡可
学号	2340850042
学习效率	3.55
当前排名	17



1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

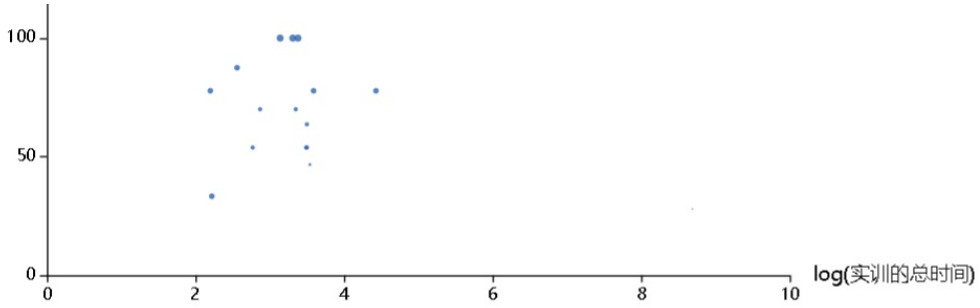
能力值

能力值（实训获得经验值/实训评测次数）



姓名	胡可
学号	2340850042
能力	233.33

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 25.33

学员代码 代码修改行数 7 src/step1/find_pos.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
  for(int i = 0;i<n;i++){
    if(data[i]==key)return i;
  }
  return -1;
 /*------------end-----------------*/
}

int main(void)
{
int a[]={99,23,5,189,-1,66};
int x;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	808.96	2024-05-21 11:48	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 查找-进阶(二分查找)

调分: 25.33

学员代码 代码修改行数 15 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数.

 /*------------begin-----------------*/
 int l,r,mid;
 int count = 0;
 l = 0,r = n-1;
 while(l<=r){
  mid = (l+r)/2;
  if(key<data[mid])r = mid-1;
  else if(key>data[mid]) l = mid+1;
  else return ++count;
  count++;
 }
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	832.65	2024-05-21 11:48	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分：26.34

学员代码 ⓘ　　代码修改行数 13　　src/step3/array_val.c

复制代码

```
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin----------*/
  for(int i = 0;i<n;i++){
    for(int j = i+1;j<n;j++){
      if(data[i]+data[j]==target)
      {
        printf("%d,%d",i,j);
        return;
      }
    }
  }
  printf("-1,-1");
  return;
  /*----------end----------*/
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	666.12	2024-05-21 11:49	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



叶思含 按时通关

学号：2340850051　　分班：未分班

截止前完成关卡：3/3　　最新完成关卡：3/3　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-21 10:53	页面停留时长	27分 9秒	3	- -	- -	78.0/100.0	良好

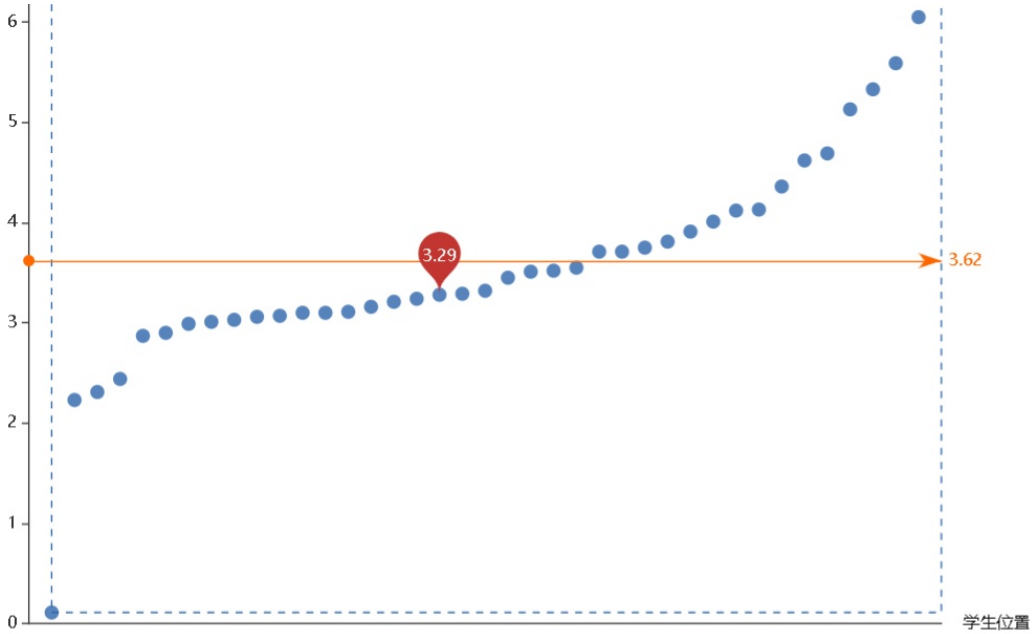
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	查找特定的数据元素-顺序查找	2024-05-21 10:25	11	1	2024-05-21 10:36	10分 44秒	否	100/100	25.33/33.33	25.33
2	查找-进阶(二分查找)	2024-05-21 10:30	22	1	2024-05-21 10:46	8分 59秒	否	300/300	25.33/33.33	25.33
3	查找进阶	2024-05-21 10:46	13	1	2024-05-21 10:53	7分 26秒	否	300/300	27.34/33.34	27.34

图形统计



姓名　叶思含
学号　2340850051
学习效率　3.29
当前排名　22

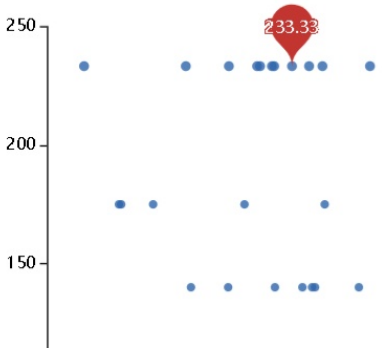


1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

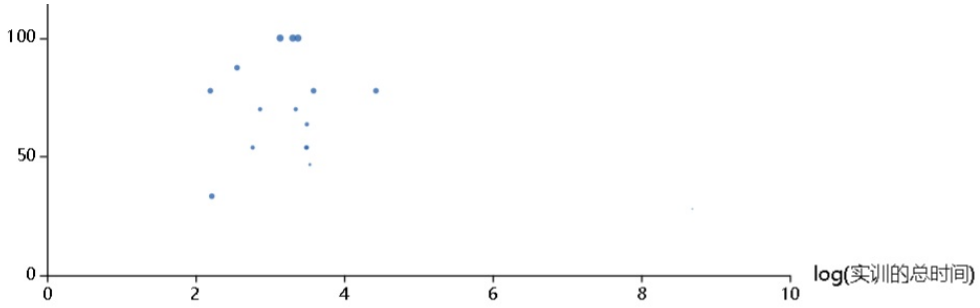
能力值

能力值（实训获得经验值/实训评测次数）

姓名	叶思含
学号	2340850051
能力	233.33



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 查找特定的数据元素-顺序查找

调分: 25.33

学员代码 代码修改行数 11 src/step1/find_pos.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key,n表示数组中数据元素的个数
 /*------------begin-----------------*/
  int i=0;
  for(i=0;i<n;i++)
  {
    if(data[i]==key)
    {
      return i;
    }
  }
  return -1;
 /*------------end-----------------*/
}
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	658.56	2024-05-21 10:36	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 查找-进阶(二分查找)

调分: 25.33

学员代码 代码修改行数 22 src/step2/find_pos2.c

复制代码

```c
#include <stdio.h>

int find_pos(int data[] ,int n, int key)
{//在数组data中查找给定数据key，n是数组中数据元素的个数，返回值是数据元素比较的次数。

 /*------------begin----------------*/
int low = 0,ans=1;
int high = n - 1;
while (low <= high)
{

  int mid = low + ((high - low) >> 1);
  if (data[mid] >key)
  {
    high = mid - 1;
    ans++;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	1069.65	2024-05-21 10:46	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第3关 查找进阶

调分: 27.34

学员代码 代码修改行数 13 src/step3/array_val.c

复制代码

```c
#include <stdio.h>
void compute(int data[],int n, int target)
{//在此处填写代码，在data数组中查找是否存在两个元素之和为target，并按要求输出
  /*----------begin----------*/
    for(int i=0;i<n-1;i++)
    {
      for(int j=i+1;j<n;j++)
      {
        if(data[i]+data[j]==target)
        {
          printf("%d,%d",i,j);
          return 0;
        }
      }

    }
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	261.24	2024-05-21 10:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-07-04 20:26	页面停留时长	27分 29秒	3	- -	- -	95.0/100.0	优秀

阶段成绩

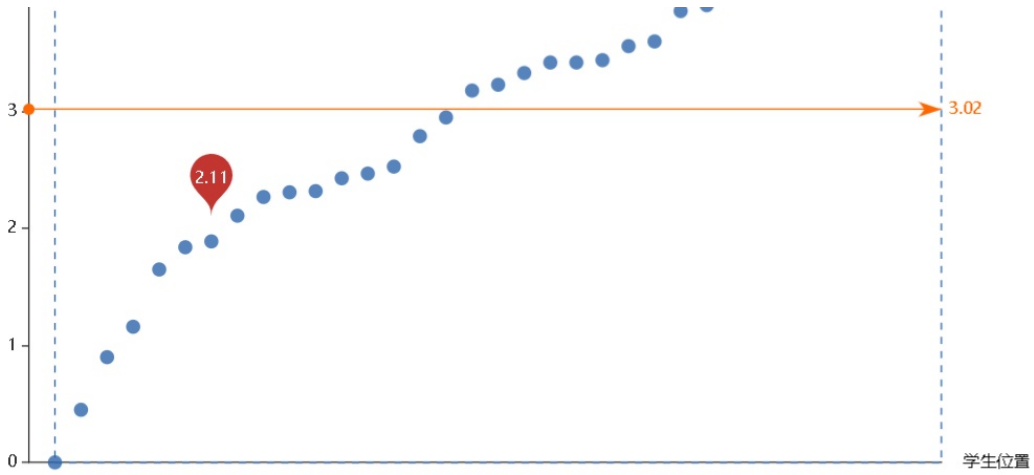
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-07-04 19:58	28	1	2024-07-04 20:12	13分 33秒	否	100/100	50.00/50	50
2	实现KMP字符串匹配	2024-07-04 20:12	17	2	2024-07-04 20:26	13分 56秒	否	100/100	45.00/50	45

图形统计



姓名	李雯君
学号	2340850043
学习效率	2.11
当前排名	28

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

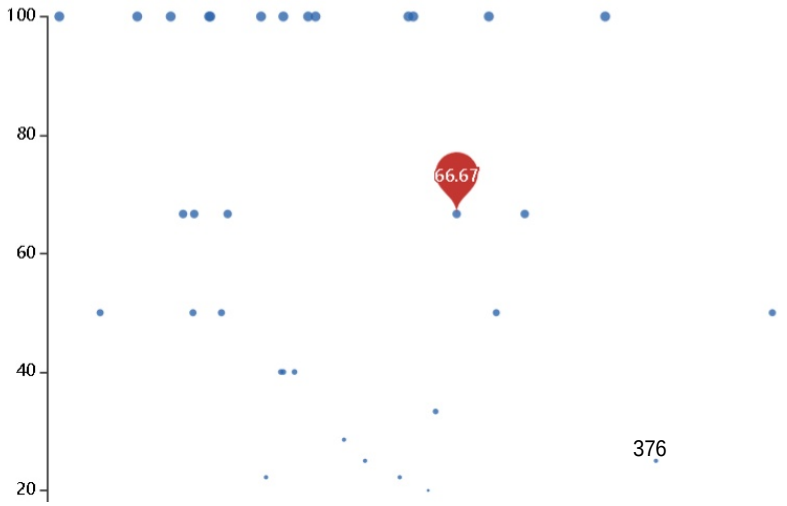


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

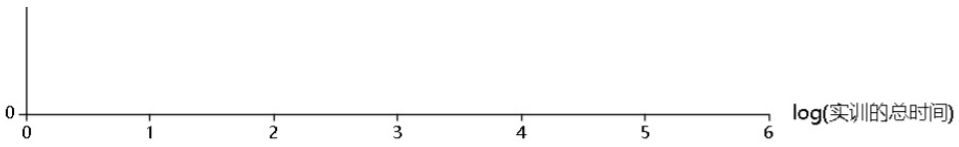
能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	66.67



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分: 50

学员代码 代码修改行数 28 step1/mystr.cpp

复制代码

```
/*************************************************************
    date: April 2009
    copyright: Zhu En
    DO NOT distribute this code.
*************************************************************/

int FindSubStr(char* t, char* p)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
{
    // 请在此添加代码，补全函数FindSubStr
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	201	2024-07-04 20:12	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 45

学员代码 代码修改行数 17 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
//
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"
#include<string.h>


void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	487.29	2024-07-04 20:13	查看代码
2	评测通过	0.03	79.56	2024-07-04 20:26	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-07-03 21:24	页面停留时长	37分 41秒	4	- -	- -	96.0/100.0	优秀

阶段成绩

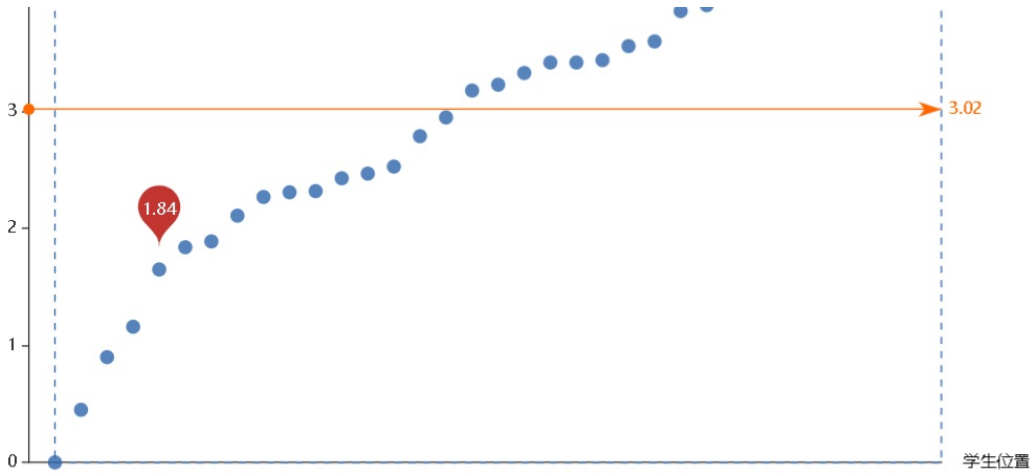
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-07-02 21:13	12	3	2024-07-03 21:22	35分 51秒	否	100/100	50.00/50	50
2	实现KMP字符串匹配	2024-07-03 21:22	12	1	2024-07-03 21:24	1分 50秒	否	100/100	46.00/50	46

图形统计



姓名	黄心媛
学号	2340850073
学习效率	1.84
当前排名	30

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

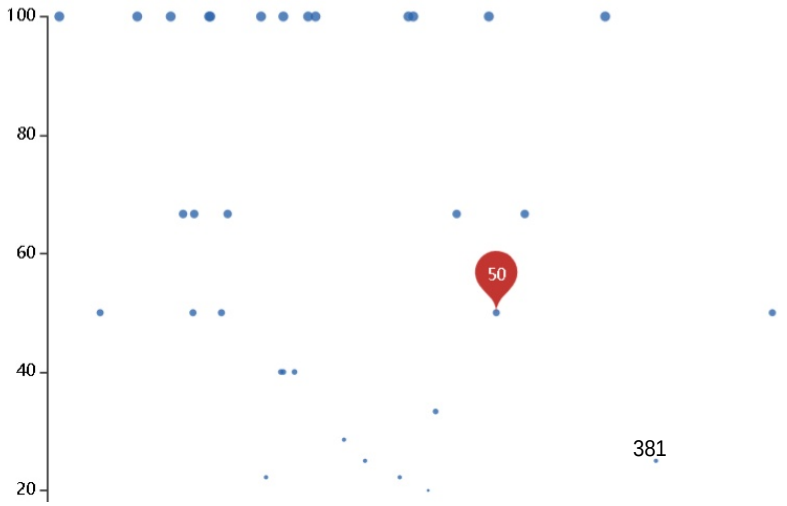


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

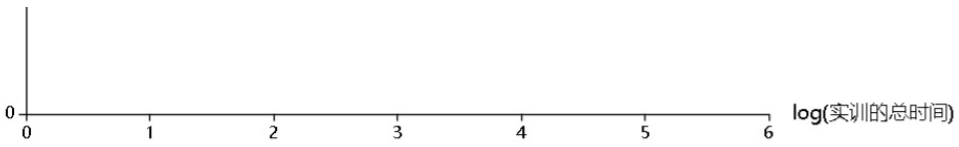
能力值

能力值（实训获得经验值/实训评测次数）

姓名	黄心媛
学号	2340850073
能力	50



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分：50

学员代码 | 代码修改行数 12 | step1/mystr.cpp

复制代码

```
/*************************************************************
    date: April 2009
    copyright: Zhu En
    DO NOT distribute this code.
*************************************************************/

int FindSubStr(char* t, char* p)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
{
    // 请在此添加代码，补全函数FindSubStr
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.07	590.58	2024-07-02 21:42	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	0	2024-07-02 21:47	查看代码
3	评测通过	0.05	455.43	2024-07-03 21:22	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 46

学员代码 代码修改行数 12 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"
////////////////////////////////////////////////////

void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
    // 请在此添加代码，补全函数KmpGenNext
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.03	220.74	2024-07-03 21:24	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



周钰欣 按时通关

学号：2340850045　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-07-05 11:03	页面停留时长	6分 45秒	5	- -	- -	88.0/100.0	良好

阶段成绩

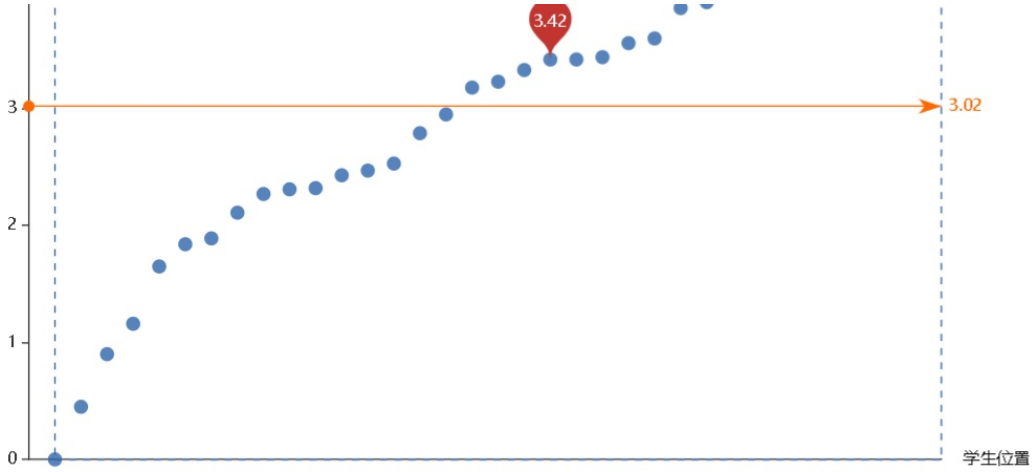
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-07-05 10:56	34	3	2024-07-05 11:00	3分 9秒	否	100/100	44.00/50	44
2	实现KMP字符串匹配	2024-07-05 11:00	17	2	2024-07-05 11:03	3分 36秒	否	100/100	44.00/50	44

图形统计



姓名	周钰欣
学号	2340850045
学习效率	3.42
当前排名	15

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

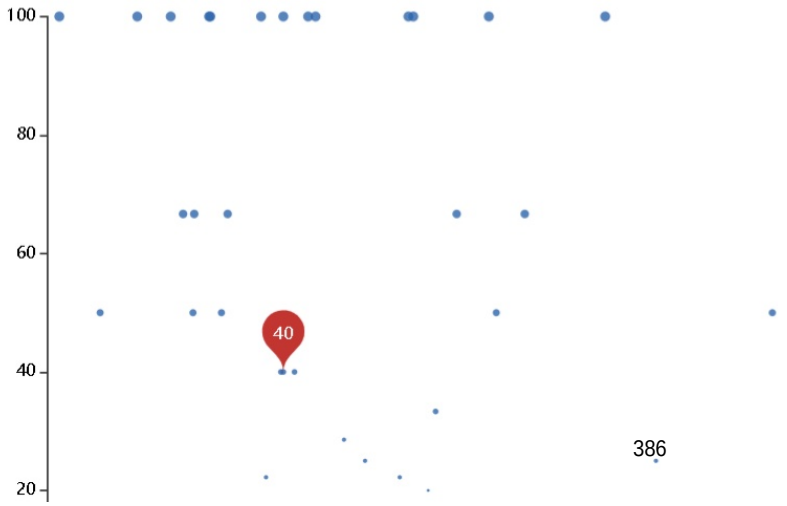


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

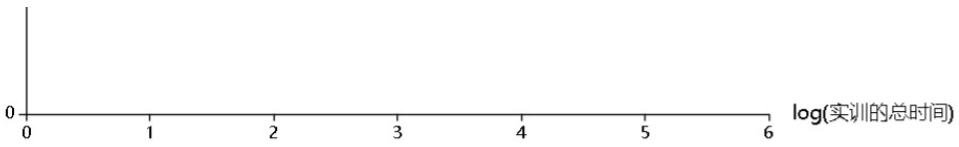
能力值

能力值（实训获得经验值/实训评测次数）

姓名	周钰欣
学号	2340850045
能力	40



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分：44

学员代码 代码修改行数 34 step1/mystr.cpp

复制代码

```
/*************************************************************
    date: April 2009
    copyright: Zhu En
    DO NOT distribute this code.
*************************************************************/
int FindSubStr(char* t, char* p)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
{
    // 请在此添加代码，补全函数FindSubStr
    /********** Begin *********/
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.03	265.98	2024-07-05 10:57	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	143.67	2024-07-05 10:59	查看代码
3	评测通过	0.03	143.67	2024-07-05 11:00	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 44

学员代码 代码修改行数 17 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"
#include <string.h>
////////////////////////////////////////////////////

void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	265.98	2024-07-05 11:02	查看代码
2	评测通过	0.01	143.67	2024-07-05 11:03	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



于利友 按时通关

学号：2340850060　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-07-05 21:52	页面停留时长	2时 16分 39秒	8	- -	- -	84.0/100.0	良好

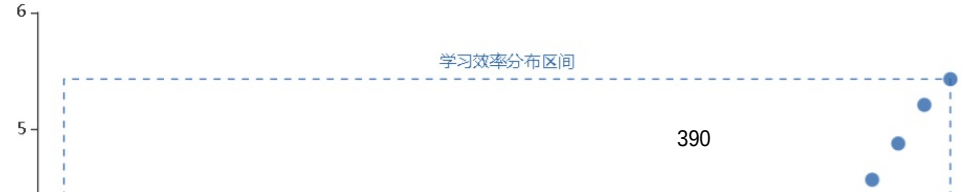
阶段成绩

关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-07-05 19:35	33	2	2024-07-05 21:23	1时 48分 7秒	否	100/100	44.00/50	44
2	实现KMP字符串匹配	2024-07-05 21:23	12	6	2024-07-05 21:52	28分 32秒	否	100/100	40.00/50	40

图形统计

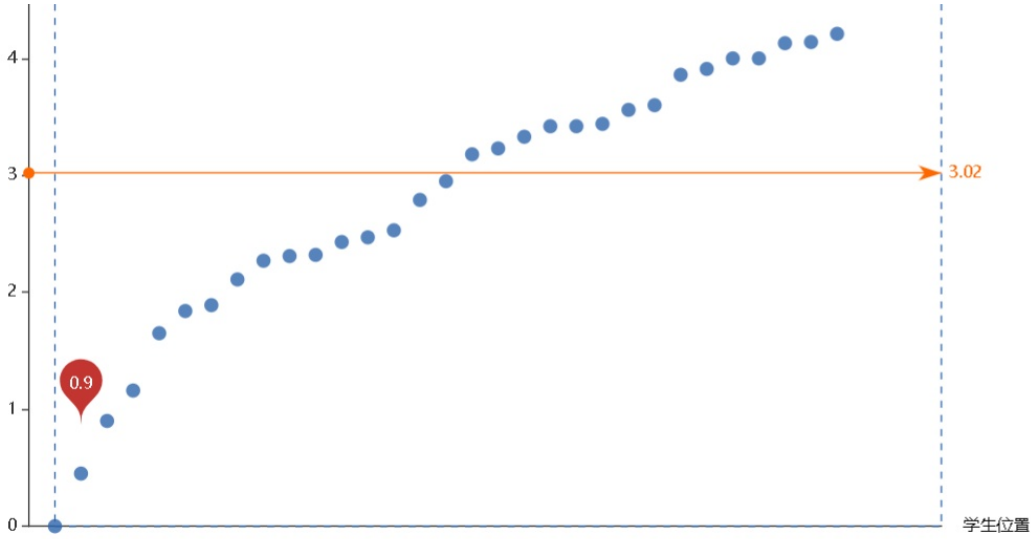
学习效率

学习效率=log(实训总得分/实训总耗时)



姓名　于利友
学号　2340850060
学习效率　0.9
当前排名　33

1个小圆点代表1个学生的学习效率



红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高
橙色横向箭头：中位值，表示整个课堂学生表现的平均值

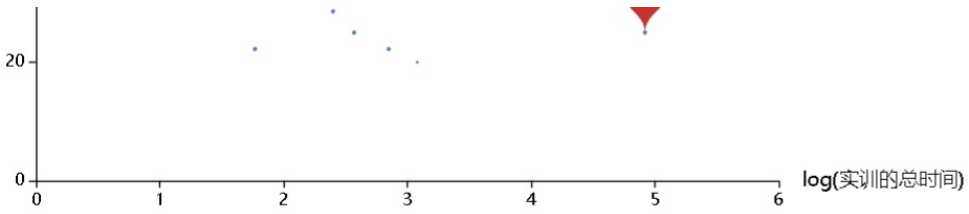
能力值

能力值（实训获得经验值/实训评测次数）

姓名	于利友
学号	2340850060
能力	25



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分：44

学员代码 代码修改行数 33 step1/mystr.cpp

复制代码

```
/*************************************************************
    date: April 2009
    copyright: Zhu En
    DO NOT distribute this code.
*************************************************************/

int FindSubStr(char* t, char* p)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
{
    // 请在此添加代码，补全函数FindSubStr
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0	464.91	2024-07-05 21:22	查看代码
2	评测通过	0.05	113.28	2024-07-05 21:23	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 40

学员代码 代码修改行数 12 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
////////////////////////////////////////////////////
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"
////////////////////////////////////////////////////

void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
    // 请在此添加代码，补全函数KmpGenNext
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有3组测试集，其中有3组测试结果不匹配	0.03	147.81	2024-07-05 21:29	查看代码
2	共有3组测试集，其中有3组测试结果不匹配	0	147.81	2024-07-05 21:29	查看代码
3	共有3组测试集，其中有3组测试结果不匹配	0.02	281.04	2024-07-05 21:36	查看代码
4	共有3组测试集，其中有3组测试结果不匹配	0.03	195.75	2024-07-05 21:42	查看代码
5	共有3组测试集，其中有3组测试结果不匹配	0	172.08	2024-07-05 21:43	查看代码
6	评测通过	0.01	195.75	2024-07-05 21:52	查看代码

代码评价 一键AI生成 编辑

总体评价

点评作业



邹璇 按时通关

学号：2340850054　　分班：未分班

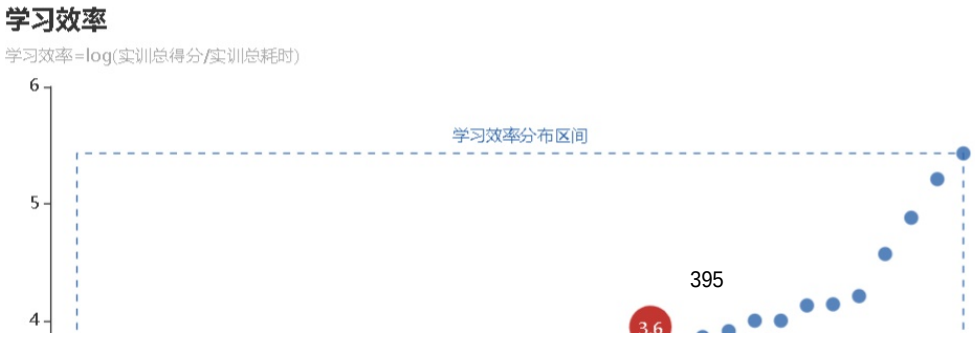
截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-23 16:53	页面停留时长	5分 37秒	2	- -	- -	75.0/100.0	良好

阶段成绩

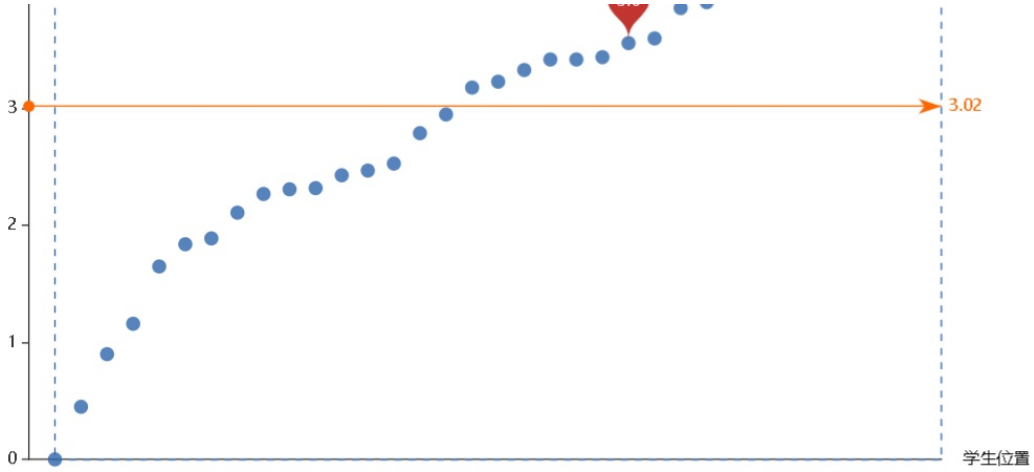
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-06-23 16:47	48	1	2024-06-23 16:48	1分 25秒	否	100/100	35.00/50	35
2	实现KMP字符串匹配	2024-06-23 16:49	24	1	2024-06-23 16:53	4分 12秒	否	100/100	40.00/50	40

图形统计



姓名　邹璇
学号　2340850054
学习效率　3.6
当前排名　12

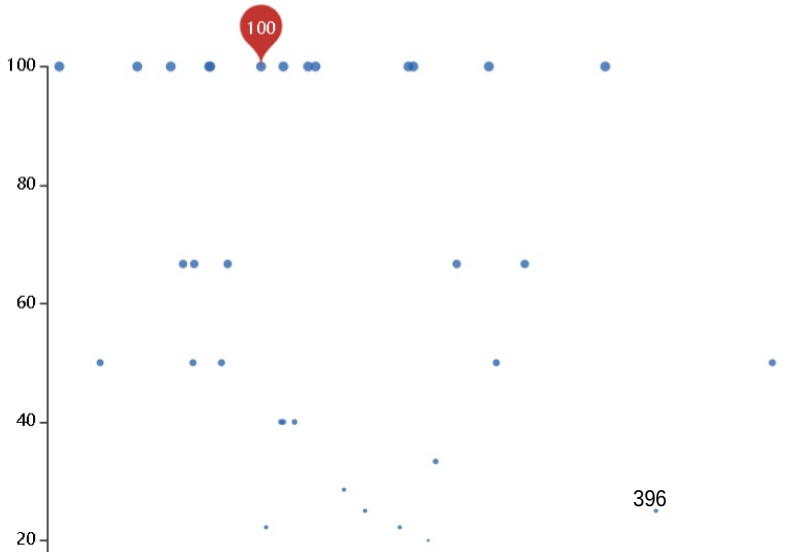
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高



橙色横向箭头：中位值，表示整个课堂学生表现的平均值

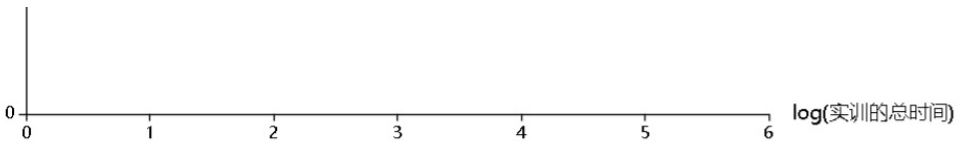
能力值

能力值（实训获得经验值/实训评测次数）



姓名	邹璇
学号	2340850054
能力	100

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分：35

学员代码 ⓘ　　代码修改行数 48　　step1/mystr.cpp

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#include "mystr.h"
#pragma warning(disable:4996)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
int FindSubStr(char* t, char* p)
{
    // 请在此添加代码，补全函数FindSubStr
    /********** Begin *********/
    int i=0,j=0;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	3660.06	2024-06-23 16:48	查看代码

代码评价　　一键AI生成　编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 40

学员代码 代码修改行数 24 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
//
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"


void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
    // 请在此添加代码，补全函数KmpGenNext
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	249.69	2024-06-23 16:53	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



张丽芳 按时通关

学号：2340850068　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-07-03 21:54	页面停留时长	19分 18秒	2	- -	- -	76.0/100.0	良好

阶段成绩

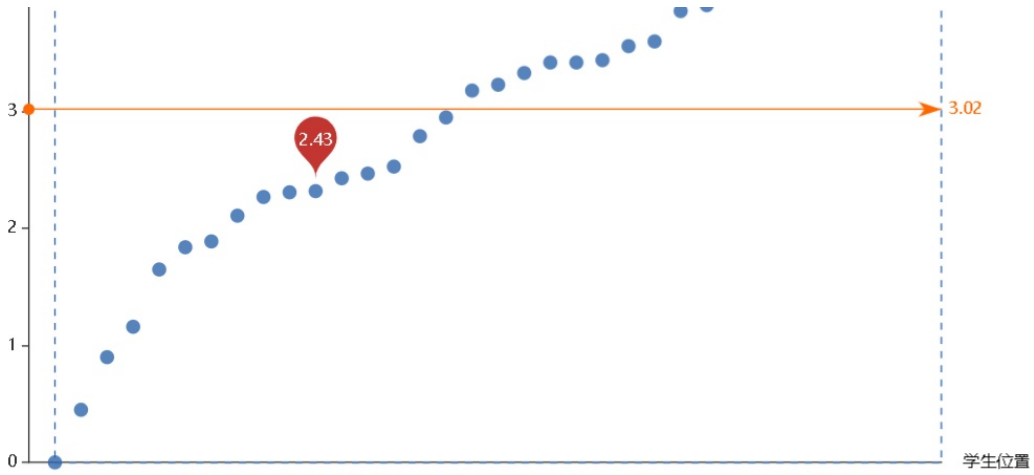
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	实现朴素的字符串匹配	2024-07-03 21:34	48	1	2024-07-03 21:52	18分 24秒	否	100/100	36.00/50	36
2	实现KMP字符串匹配	2024-07-03 21:53	24	1	2024-07-03 21:54	54秒	否	100/100	40.00/50	40

图形统计



姓名	张丽芳
学号	2340850068
学习效率	2.43
当前排名	24

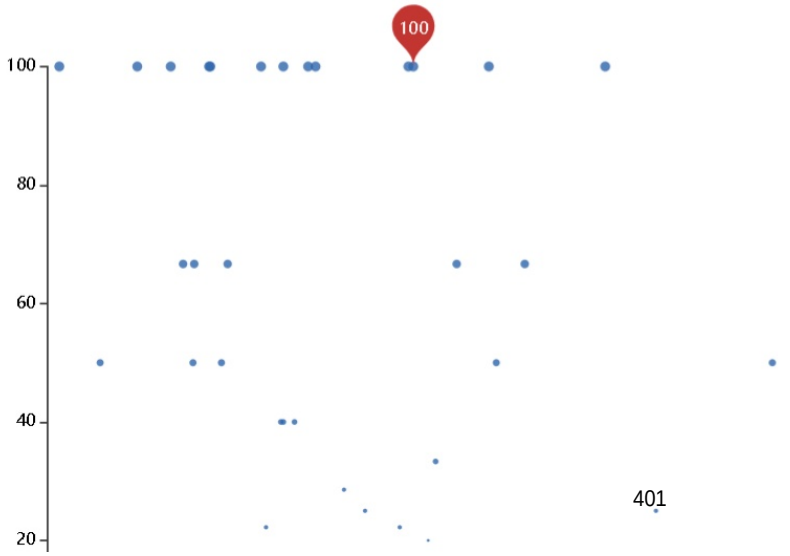
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高



橙色横向箭头：中位值，表示整个课堂学生表现的平均值

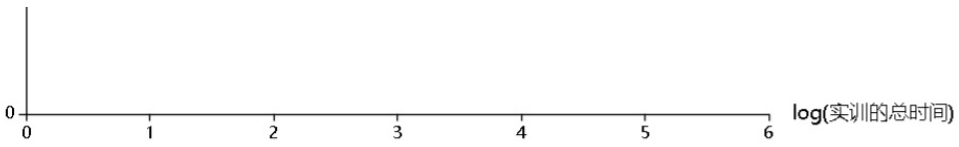
能力值

能力值（实训获得经验值/实训评测次数）



姓名	张丽芳
学号	2340850068
能力	100

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 实现朴素的字符串匹配

调分： 36

学员代码 代码修改行数 48 step1/mystr.cpp

复制代码

```
#include <stdio.h>
#include <stdlib.h>
#include "mystr.h"
#pragma warning(disable:4996)
/*
从字符串t查找子字符串p。
字符串以数值结尾，例如p="str",那么p[0]='s',p[1]='t',p[2]='r',p[3]=0。
采用朴素的匹配算法。
返回子字符串第一次出现的位置,例如t="string ring",p="ring"，则返回2。
若没有找到，则返回-1。
*/
int FindSubStr(char* t, char* p)
{
    // 请在此添加代码，补全函数FindSubStr
    /********** Begin *********/
    int i=0,j=0;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.05	191.49	2024-07-03 21:52	查看代码

代码评价 一键AI生成 编辑

暂无评价~

第2关 实现KMP字符串匹配

调分: 40

学员代码 代码修改行数 24 step2/kmp.cpp

复制代码

```
/*************************************************************
    date:
    copyright: Zhu En
    DO NOT distribute this code without my permission.
*************************************************************/
//字符串 实现文件
//
#include <stdio.h>
#include <stdlib.h>
#include "kmp.h"

void KmpGenNext(char* p, int* next)
//生成p的next数组，next数组长度大于等于字符串p的长度加1
{
    // 请在此添加代码，补全函数KmpGenNext
    /********** Begin *********/
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.04	483.84	2024-07-03 21:54	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



李雯君 按时通关

学号：2340850043　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 11:46	页面停留时长	32分 27秒	6	- -	- -	91.0/100.0	优秀

阶段成绩

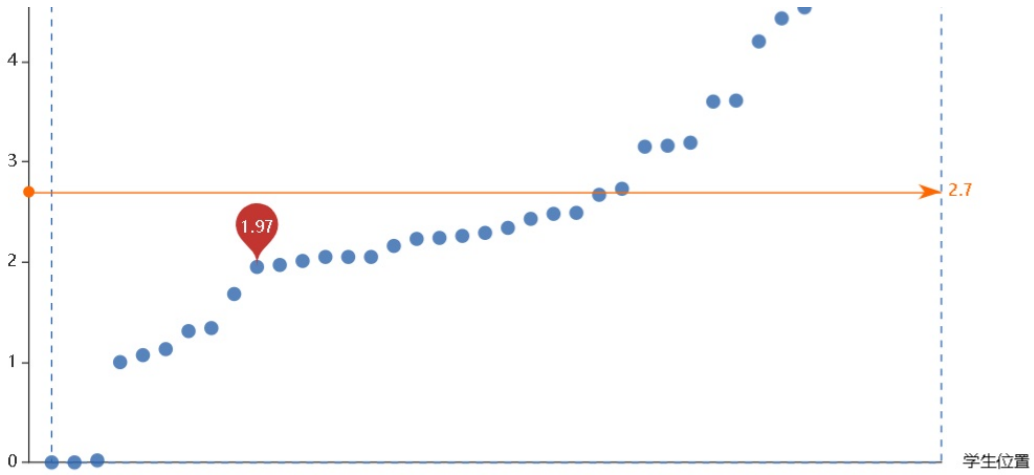
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-05-28 10:20	25	5	2024-06-04 11:46	31分 9秒	否	100/100	45.00/50	45
2	堆排序	2024-05-28 10:35	46	1	2024-06-04 11:46	1分 18秒	否	100/100	46.00/50	46

图形统计



姓名　李雯君
学号　2340850043
学习效率　1.97
当前排名　30

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

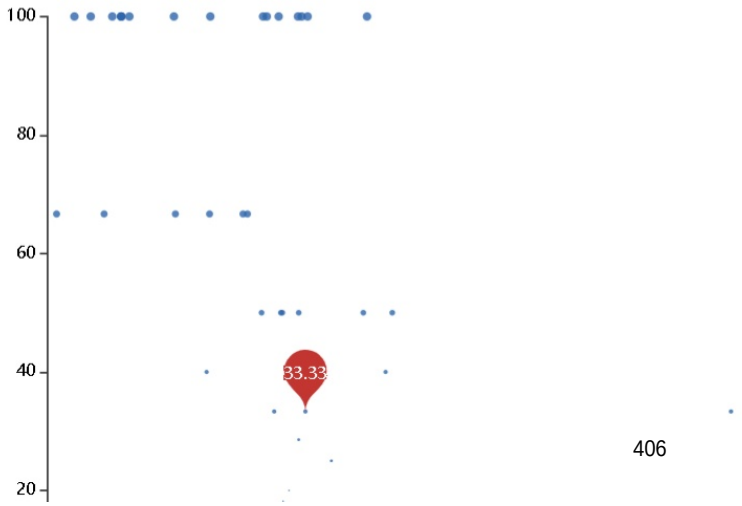


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

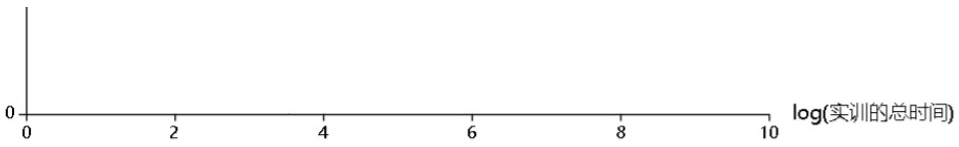
能力值

能力值（实训获得经验值/实训评测次数）

姓名	李雯君
学号	2340850043
能力	33.33



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分：45

学员代码 代码修改行数 25 step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{

  for(int i = 0;i<n-1;i++)
  {
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	2440.24	2024-05-28 10:40	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.01	493.9	2024-05-28 10:40	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	467.96	2024-06-04 11:42	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0	471.98	2024-06-04 11:44	查看代码
5	评测通过	0.02	465.08	2024-06-04 11:46	查看代码

代码评价

一键AI生成 编辑

408

暂无评价~

第2关 堆排序

调分：46

学员代码 | 代码修改行数 46 | step2/step2.c

复制代码

```c
#include<stdio.h>

int size;

void print(int a[] ,int n)
{
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void swap(int *i,int *j){
    int temp = *i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	116.86	2024-06-04 11:46	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



黄心媛 按时通关

学号：2340850073　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-08 21:24	页面停留时长	45分 51秒	8	- -	- -	92.0/100.0	优秀

阶段成绩

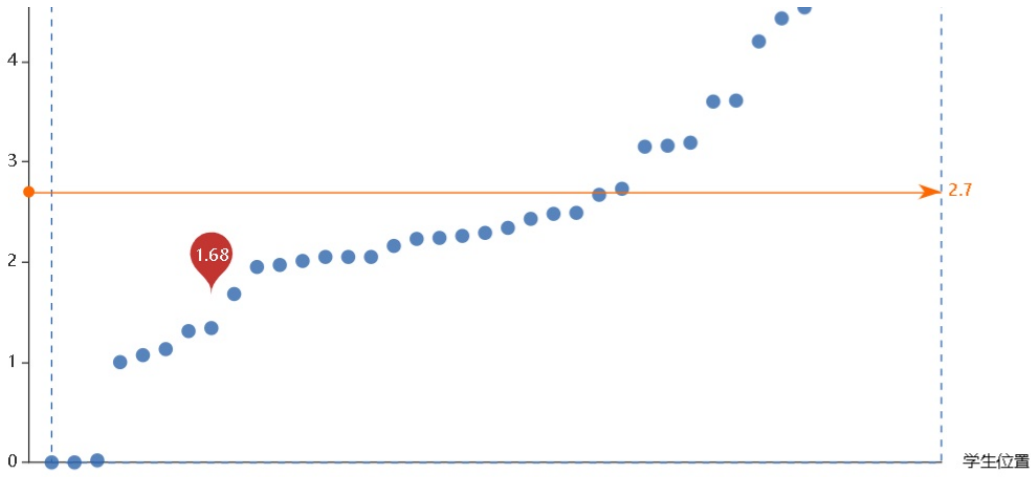
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-06-05 22:01	17	5	2024-06-07 22:03	31分 10秒	否	100/100	45.00/50	45
2	堆排序	2024-06-07 22:03	35	3	2024-06-08 21:24	14分 41秒	否	100/100	47.00/50	47

图形统计



姓名　黄心媛
学号　2340850073
学习效率　1.68
当前排名　32

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

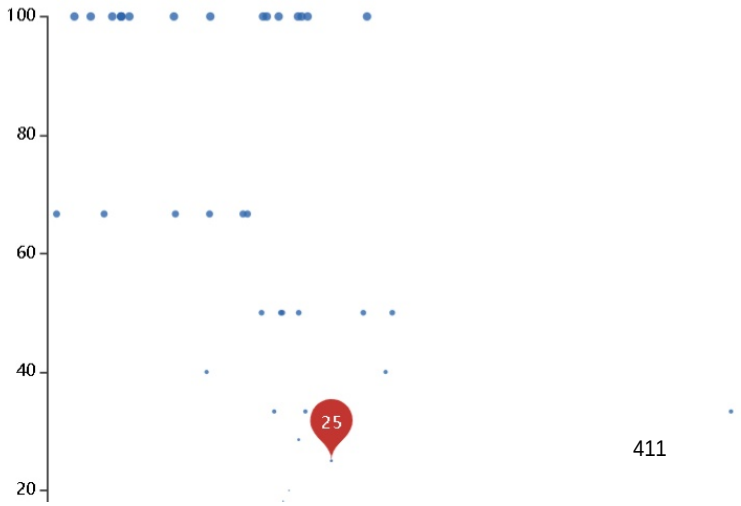


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

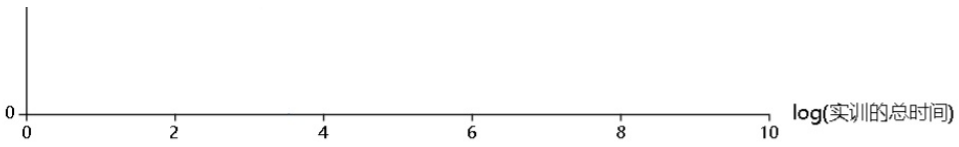
能力值

能力值（实训获得经验值/实训评测次数）

姓名	黄心媛
学号	2340850073
能力	25



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分：45

学员代码 代码修改行数 17 step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
    for(int i=0;i<n;i++)
        printf("%d ",a[i]);
        printf("\n");


}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{
    int i,j,t,s;
    for(i=0;i<n-1;i++)
    {
        s=i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	2439.72	2024-06-05 22:08	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0.02	74.36	2024-06-07 21:42	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0.02	315.82	2024-06-07 21:50	查看代码
4	共有2组测试集，其中有2组测试结果不匹配	0.01	321.48	2024-06-07 21:52	查看代码
5	评测通过	0.02	103.88	2024-06-07 22:03	查看代码

代码评价

一键AI生成 编辑

413

暂无评价~

第2关 堆排序

调分：47

学员代码 ⓘ　　代码修改行数 35　　step2/step2.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
    for(int i=1;i<=n;i++)
        printf("%d ",a[i]);
        printf("\n");
}

void sift(int a[],int k,int m)
{
    int i,j,t;
    i=k;j=2*i;
    while(j<=m)
    {
        if(j<m&&a[j]<a[j+1])
            j++;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.02	299.42	2024-06-08 21:23	查看代码
2	评测通过	0.01	1332.38	2024-06-08 21:24	查看代码
3	评测通过	0.02	317.9	2024-06-08 21:29	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



张佳慧 按时通关

学号：2340850044　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-28 10:41	页面停留时长	6天 22时 44分 35秒	6	- -	- -	86.0/100.0	良好

阶段成绩

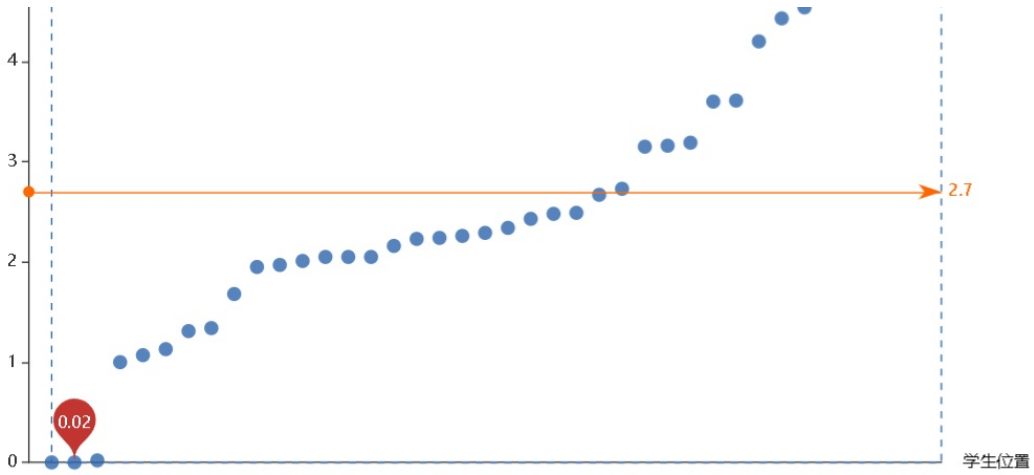
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-05-21 11:56	20	2	2024-05-28 10:25	6天 22时 28分 37秒	否	100/100	44.00/50	44
2	堆排序	2024-05-28 10:25	42	4	2024-05-28 10:41	15分 58秒	否	100/100	42.00/50	42

图形统计



姓名　张佳慧
学号　2340850044
学习效率　0.02
当前排名　38

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高



橙色横向箭头：中位值，表示整个课堂学生表现的平均值

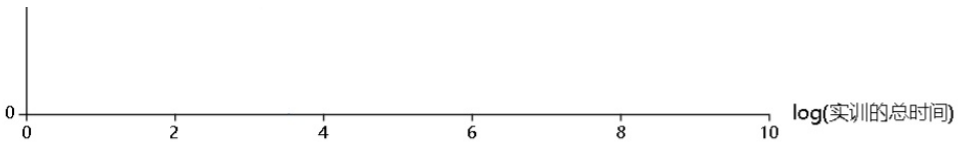
能力值

能力值（实训获得经验值/实训评测次数）

姓名	张佳慧
学号	2340850044
能力	33.33



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分: 44

学员代码 代码修改行数 20 step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i=0;
  while(i<n)
  {
    printf("%d ",a[i]);
    i++;
  }
  printf("\n");
}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{
  for(int i=0;i<n-1;i++)
  {
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.01	2439.9	2024-05-28 10:24	查看代码
2	评测通过	0.01	444.54	2024-05-28 10:25	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 堆排序

调分: 42

学员代码 | 代码修改行数 42 | step2/step2.c

复制代码

```c
#include <stdio.h>

int size;

void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i=0;
  while(i<n)
  {
    printf("%d ",a[i]);
    i++;
  }
  printf("\n");
}

void swap(int *i,int *j)
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	625.1	2024-05-28 10:37	查看代码
2	共有2组测试集，其中有2组测试结果不匹配	0	2439.9	2024-05-28 10:38	查看代码
3	共有2组测试集，其中有2组测试结果不匹配	0	479.1	2024-05-28 10:40	查看代码
4	评测通过	0.01	632.08	2024-05-28 10:41	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



马文韬 按时通关

学号：2340850067　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 13:00	页面停留时长	38秒	2	- -	- -	80.0/100.0	良好

阶段成绩

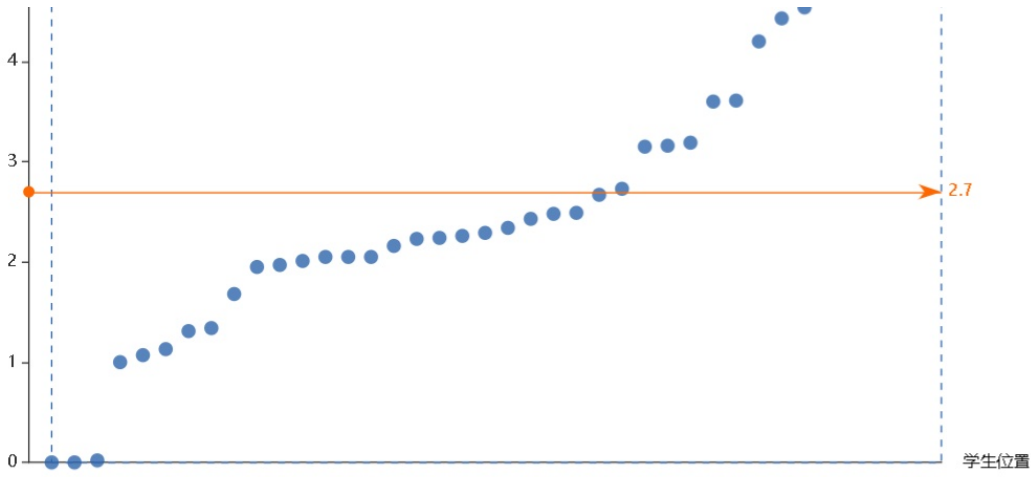
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-06-04 12:59	22	1	2024-06-04 12:59	22秒	否	100/100	40.00/50	40
2	堆排序	2024-06-04 12:59	46	1	2024-06-04 13:00	16秒	否	100/100	40.00/50	40

图形统计



姓名　马文韬
学号　2340850067
学习效率　5.76
当前排名　2

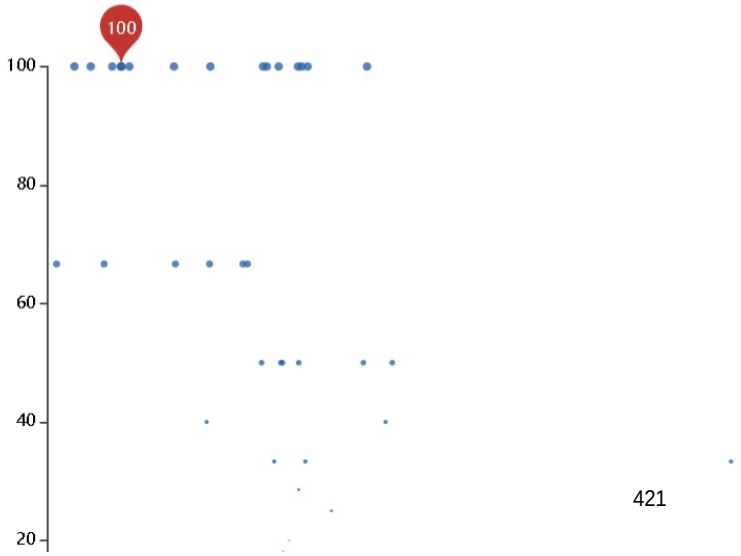
1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高



橙色横向箭头：中位值，表示整个课堂学生表现的平均值

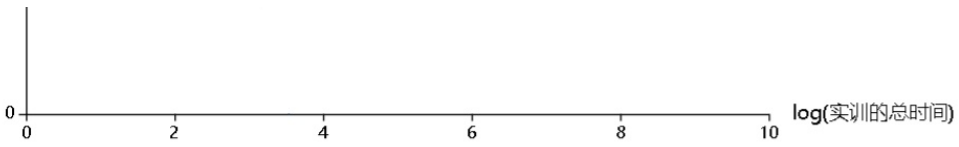
能力值

能力值（实训获得经验值/实训评测次数）



姓名	马文韬
学号	2340850067
能力	100

1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分: 40

学员代码 代码修改行数 22 step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{

    for(int i = 0;i<n-1;i++){
        int index = i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	2439.28	2024-06-04 12:59	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 堆排序

调分: 40

学员代码 ⓘ　　代码修改行数 46　　step2/step2.c

复制代码

```c
#include<stdio.h>

int size;

void print(int a[] ,int n)
{
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void swap(int *i,int *j){
    int temp = *i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.01	2439.28	2024-06-04 13:00	查看代码

代码评价

一键AI生成　编辑

暂无评价~

总体评价

点评作业



钱杨 按时通关

学号：2340850059　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-06-04 10:37	页面停留时长	1时 10分 29秒	4	- -	- -	78.0/100.0	良好

阶段成绩

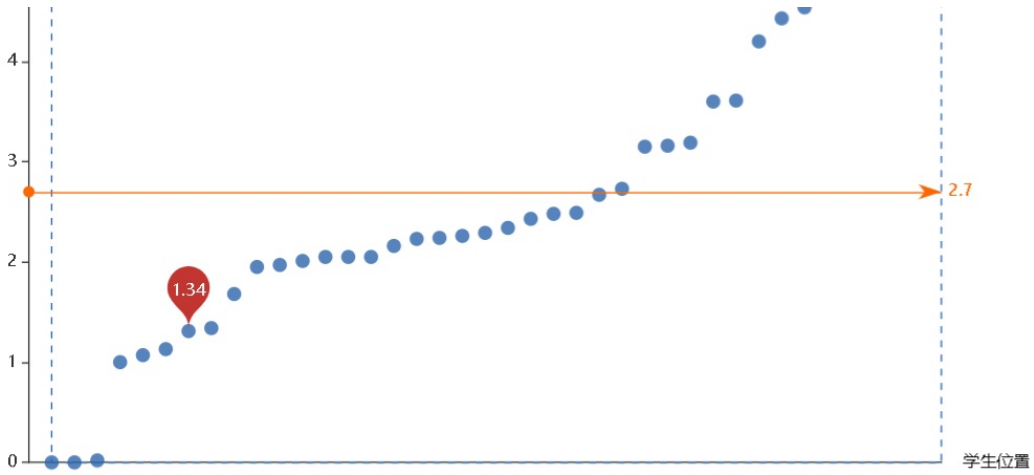
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-05-28 10:22	22	1	2024-06-04 10:31	39分 37秒	否	100/100	38.00/50	38
2	堆排序	2024-05-28 10:42	47	3	2024-06-04 10:37	30分 52秒	否	100/100	40.00/50	40

图形统计



姓名　钱杨
学号　2340850059
学习效率　1.34
当前排名　33

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

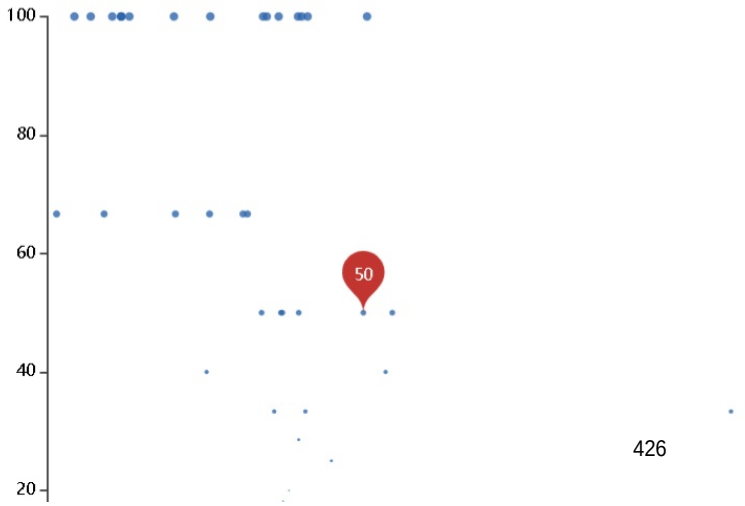


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

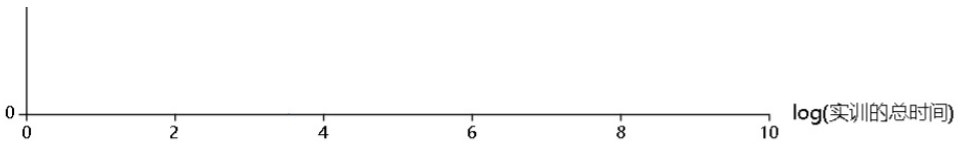
能力值

能力值（实训获得经验值/实训评测次数）

姓名	钱杨
学号	2340850059
能力	50



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分：38

学员代码 ⓘ　　代码修改行数 22　　step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{

  for(int i = 0;i<n-1;i++){
      int index = i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	156.88	2024-06-04 10:31	查看代码

代码评价

一键AI生成　编辑

暂无评价~

第2关 堆排序

调分：40

学员代码 | 代码修改行数 47 | step2/step2.c

复制代码

```c
#include<stdio.h>

int size;

void print(int a[] ,int n)
{
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void swap(int *i,int *j){
    int temp = *i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0.03	103.18	2024-06-04 10:30	查看代码
2	评测通过	0.02	312.34	2024-06-04 10:37	查看代码
3	评测通过	0.02	327.8	2024-06-04 10:38	查看代码

代码评价

一键AI生成 编辑

暂无评价~

总体评价

点评作业



王颖 按时通关

学号：2340850062　　分班：未分班

截止前完成关卡：2/2　　最新完成关卡：2/2　　完成效率：--　　课堂最高完成效率：--

通关时间	计时规则	实训总耗时	评测次数	查重扣分	补交扣分	最终成绩	总评
2024-05-28 11:30	页面停留时长	14分 52秒	3	- -	- -	76.0/100.0	良好

阶段成绩

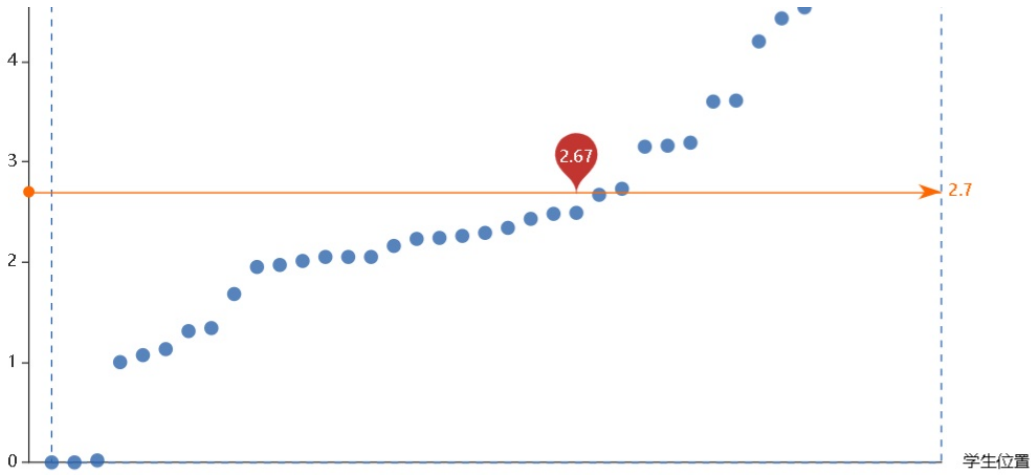
关卡	任务名称	开启时间	代码修改行数	评测次数	完成时间	实训耗时	是否查看答案	经验值	关卡得分	调分
1	直接选择排序	2024-05-28 11:17	22	2	2024-05-28 11:25	8分 49秒	否	100/100	38.00/50	38
2	堆排序	2024-05-28 11:17	46	1	2024-05-28 11:30	6分 3秒	否	100/100	38.00/50	38

图形统计



姓名　王颖
学号　2340850062
学习效率　2.67
当前排名　16

1个小圆点代表1个学生的学习效率
红色水滴表示当前学生的学习效率值
小圆点越高，学习效率越高

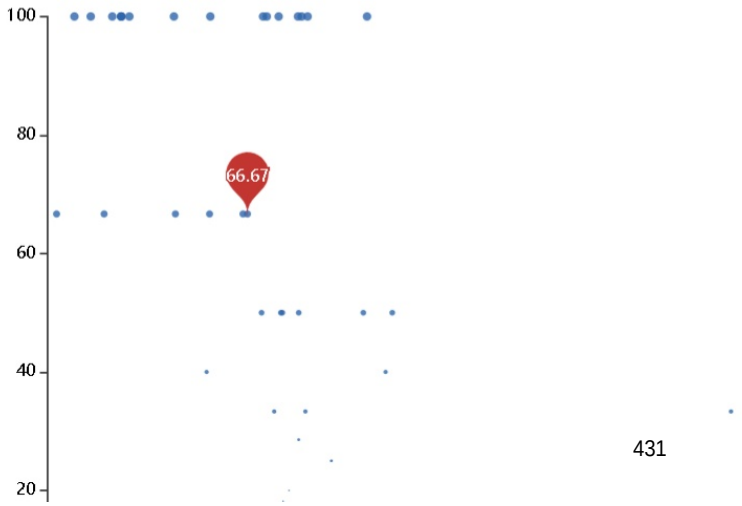


橙色横向箭头：中位值，表示整个课堂学生表现的平均值

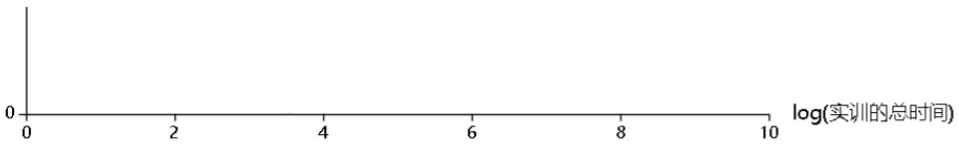
能力值

能力值（实训获得经验值/实训评测次数）

姓名	王颖
学号	2340850062
能力	66.67



1个小圆点代表1个学生的能力值
红色水滴表示当前学生的能力值
小圆圈越大，评测次数越少



实训详情

第1关 直接选择排序

调分：38

学员代码 代码修改行数 22 step1/step1.c

复制代码

```c
#include <stdio.h>
void print(int a[] ,int n)
{//输出数组元素，数字之间以一个空格为界，输出结束后换一行
  int i = 0;
  while(i<n){
    printf("%d ",a[i]);
      i++;
  }
  printf("\n");
}

void selectSort(int a[] ,int n) //对数组a中的n个元素进行排序，调用print函数输出每趟排序后的结果
{

  for(int i = 0;i<n-1;i++){
      int index = i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	共有2组测试集，其中有2组测试结果不匹配	0	2439.98	2024-05-28 11:24	查看代码
2	评测通过	0.01	2439.6	2024-05-28 11:25	查看代码

代码评价

一键AI生成 编辑

暂无评价~

第2关 堆排序

调分：38

学员代码 ⓘ　　代码修改行数 46　　step2/step2.c

复制代码

```c
#include<stdio.h>

int size;

void print(int a[] ,int n)
{
    int i = 0;
    while(i<n){
        printf("%d ",a[i]);
            i++;
    }
    printf("\n");
}

void swap(int *i,int *j){
    int temp = *i;
```

评测次数	详细信息	评测耗时(S)	内存消耗(MB)	评测时间	操作
1	评测通过	0.02	2439.98	2024-05-28 11:30	查看代码

代码评价

一键AI生成　编辑

暂无评价~

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

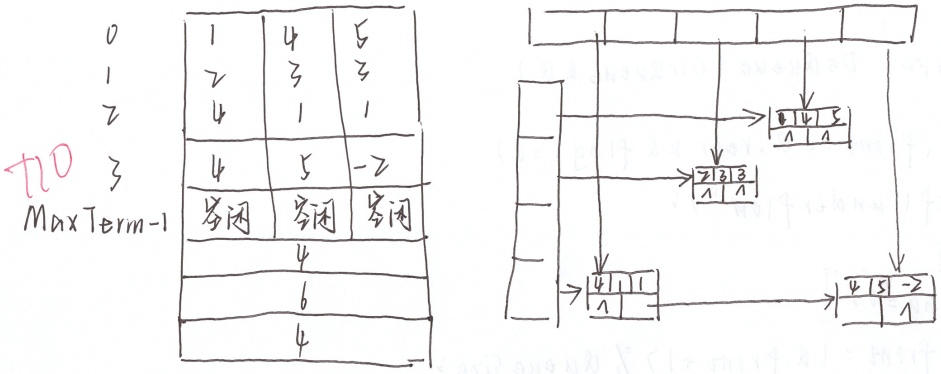
姓名[illegible]　计算机工程学院　2023　级　计科　专业　2　班　学号：2340850068

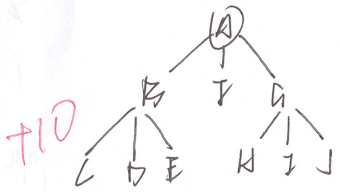
题号	一	二	三	总分
得分	24	40	28	92

得 分 24　一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	C	B	C	B	D	C	D	B
11	12	13	14	15					
A	C	D	C	C A					

得 分 40　二、 结构分析题（每题 10 分，共 40 分）





+10



3. ① 0, 01, 010, 0101

+10 ② 1, 00, 011, 0100

4. $n_0 = n - n_k = n - \frac{n-1}{k}$

+10 $n_0 = \frac{n(k-1)+1}{k}$

2. int Like(BiNode *s, BiNode *t)
{
  int same;
  if(s==NULL&&t==NULL)
    return 1;
  else if((s==NULL && t!=NULL)||(s!=NULL&&t==NULL))
    return 0;
  else {
    same=Like(s->lchild, t->lchild);
    if(same==1) same=Like(s->rchild, t->rchild);
    return same;
  }
}

3. void EnQueue(CirQueue &Q, DataType x)
{
  if(Q.front==Q.rear && flag==1)
    printf("overflow");
  else {
    flag=1;
    Q.rear=(Q.rear+1)%QueueSize;
    Q.data[Q.rear]=x;
  }
}
DataType DeQueue(CirQueue &Q)
{
  if(Q.front==Q.rear && flag==0)
    printf("underflow");
  else {
    flag=0;
    Q.front=(Q.front+1)%QueueSize;
    return Q.data[Q.front];
  }
}

得分 28

三、编程题（每题 10 分，共 30 分）

1. typedef int ElemType;
typedef struct LNode {
ElemType data;
struct Lnode *Link;
}*LinkList;
int Search_k(LinkList list, int k){
LinkList p=list->link, q=list->link;
int count=0;
while(p!=NULL){
if(count<k)
  count++;
else
  q=q->link;
  p=p->link;
}
if(count<k)
  return 0;
else {
printf("%d", q->data);
return 1;
}

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

姓名 邓佳雨 计算机工程学院 2023 级 计科 专业 3 班 学号：2340850079

题号	一	二	三	总分
得分	26	40	25	91

得 分	26

一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	D	B	C	B	B	C	D	A
11	12	13	14	15					
A	~~BC~~	D	C	~~B~~					

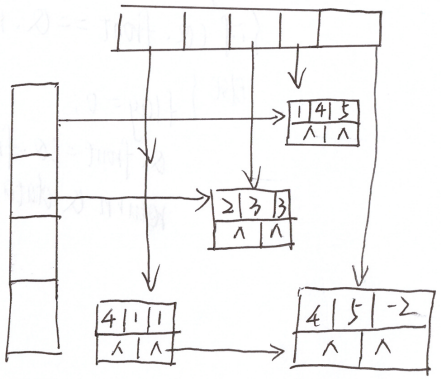
得 分	40

二、 结构分析题（每题 10 分，共 40 分）

1.

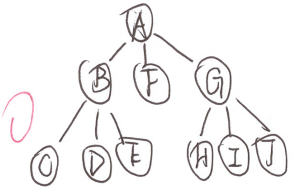
+10

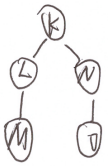
0	1	4	5
1	2	3	3
2	4	1	1
3	4	5	-2
MaxTerm-1	空闲	空闲	空闲
	4		
	6		
	4		



2.

+10





3. 1. 0, 01, 010 和以0101为前缀的编码

+10 2. 以1, 00, 011, 0100为前缀的编码.

4. $n = n_0 + n_k$.

$n = B+1 = k \times n_k + 1$

+10 $n_0 = n - n_k$

$= n - \frac{n-1}{k}$

$= \frac{n(k-1)+1}{k}$

得分	25

三、编程题（每题 10 分，共 30 分）

1.
```
typedef int ElemType;
typedef struct LNode{
  ElemType data;
  Struct Lnode * Link;} * LinkList;
int Search_k(LinkList list, int k){
  LinkList p=list->link, q=list->link;
  int count=0;
while(p!=NULL){
  if(count<k) count++;
  else q=q->link; p=p->link;}
  if(count<k) return 0;
  else{
    printf("%d", q->data);
    return 1;}}
```

+10

2.
```
{int same;
 if(s==NULL && t==NULL) return 1;
 else if((s==NULL && t!=NULL) || (s!=NULL && t==NULL))
   return 0;
 else{
   same=Like(s->lchild, t->lchild);
   if(same==1) same=Like(s->rchild, t->rchild);
   return same;}}
```

+7

3.
```
{if(Q.front==Q.rear && flag==1)
   printf("overflow");
 else{
   flag=1;
   Q.rear=(Q.rear+1)%QueueSize;
   Q.data[Q.rear]=x;}}
```

+8

```
DataType DeQueue(CirQueue &Q)
 {if(Q.front==Q.rear && flag!=0) printf("underflow");
  else{flag=0;
       Q.front=(Q.front+1)%QueueSize;
       return Q.data[Q.front];}}
```

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚 审题人：王莉娜 试卷编号：第 A 套

姓名 周雯浩 计算机工程学院 2023 级 计科 专业 2 班 学号：2340850047

题号	一	二	三	总分
得分	28	40	20	88

得 分	28

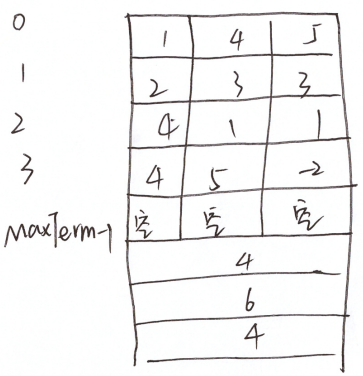
一、 选择题（每题 2 分，共 30 分）

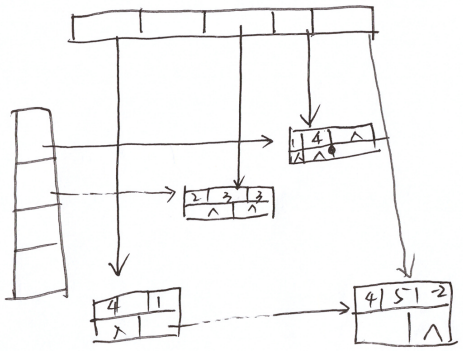
1	2	3	4	5	6	7	8	9	10
B	B	D	A	C	B	B	C	D	A
11	12	13	14	15					
A	C	D	C	C					

得 分	

二、 结构分析题（每题 10 分，共 40 分）

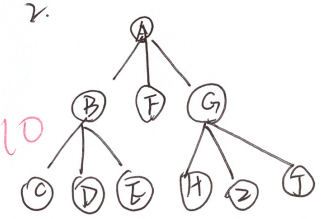
+10

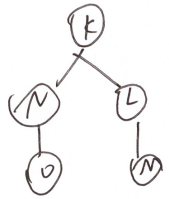




2.

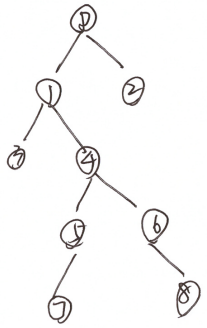
+10





3.

+10



1. 以 0, 01, 010 和 0101 为前缀

2、 以 1, 00, 011, 0100 为前缀

4. $n = n_0 + n_k$

$n = B+1 = k \cdot n_k + 1$

+10

$n_0 = n - k = n - \frac{n-1}{k} = \frac{n(k-1)+1}{k}$

3. void EnQueue(CirQueue &Q, DataType x)

{ if (Q.front == Q.rear && flag == 1) printf("overflow");

else { flag = 1;

Q.rear = (Q.rear + 1) % QueueSize;

Q.data[Q.rear] = x;

}}

DataType DeQueue(CirQueue &Q)

{ if (Q.front == Q.rear && flag == 0)

printf("underflow");

+10 else { flag = 0;

Q.front = (Q.front + 1) % QueueSize

return Q.data[Q.front];

}}

得 分 20

三、编程题（每题 10 分，共 30 分）

1、

2. int Like(BiNode *s, BiNode *t)

{ int same;

if(s == NULL && t == NULL) return 1

else if((s == NULL && t != NULL) || (s != NULL && t == NULL))

return 0;

+10 else { same = Like(s->lchild, t->lchild);

if(same == 1) same = Like(s->rchild, t->rchild);

return same;

}

}

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

姓名金佳雯　计算机工程学院　2023　级　计科　专业　3　班　学号：2244850083

题号	一	二	三	总分
得分	24	28	30	82

得分 24　一、 选择题（每题 2 分，共 30 分）

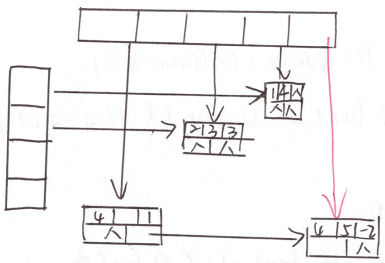
1	2	3	4	5	6	7	8	9	10
B	B	D	B	C	B	C	C	D	A
11	12	13	14	15					
A	C	D	C	D					

得分 28　二、 结构分析题（每题 10 分，共 40 分）

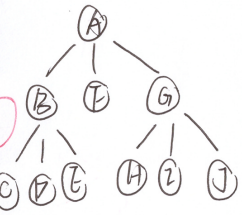
1. 三元组顺序表　十字链表

0	1	4	5
1	2	3	3
2	4	1	1
3	4	5	-2
MaxTerm-1	空闲	空闲	空闲
	4		
	6		
	4		

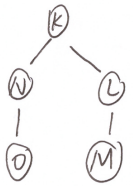
+8



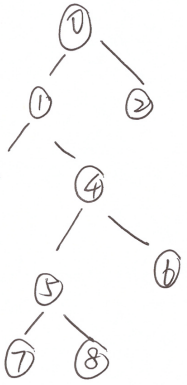
2.



+10



3.



+5

4.

~~A0~~ $n = n_0 + n_k$

$n = kn_k$

+5 $n_0 = n - n_k = n - \frac{n-1}{k}$

得分	30

三、编程题（每题 10 分，共 30 分）

1.
```
typedef int Elem Type;
typedef strust LNode {
Elem TfType data;
Struct Lnode * Link;
}* LinkList;
int Search_k(LinkList list, int k){
LinkList p=list->link, q=list->link;
int count=0;
while(p!=NULL){
if(count<k) count++;
else q=q->link; p=p->link;
}
if(count<k) return 0;
else {
printf("%d", q->data);
return 1;
}
}
```

+10

2.
```
int Like(BiNode *s, BiNode *t)
{ int same;
  if(s==NULL && t=NULL) return 1;
  else if((s== NULL && t!=NULL)||(s!=NULL && t==NULL))
      return 0;
  else {
     same = Like(s->lchild, t->lchild);
     if(same==1) same = Like(s->rchild, t->rchild);
     return same;
  }
}
```

+10

3.
```
void EnQueue(CirQueue &Q, DataType x)
{ if(Q.front==Q.rear && flag==1) printf("overflow");
  else {
    flag=1;
    Q.rear=(Q.rear+1)%QueueSize;
    Q.data[Q.rear]=x;
  }
}
DataType DeQueue(CirQueue &Q)
{ if &(Q.front==Q.rear && flag==0) printf("underflow");
  else {
    flag=0;
    Q.front=(Q.front+1)%QueueSize;
    return Q.data[Q.front];
  }
}
```

+10

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

姓名吴天润　计算机工程学院　2023　级 计科　专业　3　班　学号：2340850087

题号	一	二	三	总分
得分	24	30	24	78

得分 24　一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	D	B	C	A	B	C	D	D
11	12	13	14	15					
A	C	D	D	C					

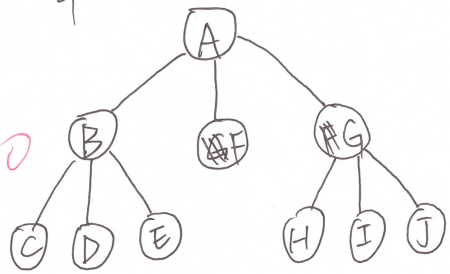
得分 30　二、 结构分析题（每题 10 分，共 40 分）

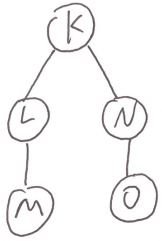
1. 三元：

1	4	5	0
2	3	3	1
4	1	1	2
4	5	-2	3
空闲	空闲	空闲	
4			
6			
4			

十字：

2.

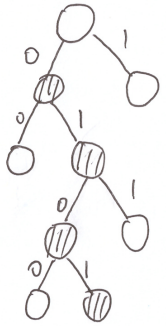




3.

编码不可能：0，01，010，0101 开头的编码是不可能的.

编码一定会有：1，00，011，0100 开头的编码是一定会有的.



分析：根据字符的编码可以画出编码的树如左边所示.

4.

$n = n_1 + n_2 + n_0$

$n_0 = n - n_1 - n_2$

$n = n_0 + n_k$

$n = k \times n_k + 1 \Rightarrow n_k = \frac{1}{k}(n+1)$

$n_0 = n - n_k = n - \frac{1}{k}(n+1)$

$= \frac{1}{k}[n(k-1)+1]$

得分	24

三、编程题（每题 10 分，共 30 分）

1.
```
typedef  int  ElemType;
typedef  struct  LNode
ElemType  LNode{
 ElemType  data;
 Struct  Lnode *Link;
}* LinkList
int  Search_k(LinkList  list, ink)

LinkList p = list→link , q=list→link;
 int count=0;
 while (p != NULL) {
    if (count <k)
      count++;
    else
      q= q→link;
      p=p→link;
 }.
```

```
if (count<k)
    return 0;
}.
```

+10

2.
```
int  like(Node *s, Node *t)
{
   int   lchild, rchild;
   if (t = NULL && s != NULL)
   return 0;
   if (t!= NULL && s= NULL)
   return 0;
   if (t→lchild = s→lichild){
       if(t→rchild = s→rchild)
           return 1;
       else
           return 0;
   }.
}.
```

+8

3.
```
void  EnQueue (cirQueue &Q, DataTypex)
{   int flag=0
    if ( Q.front =rear)
else { flag=1;.
     Q.rear=(Q.rear+1)%QueueSize;
     Q.data[Q.rear]=X;
    }
}
```

+6

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

姓名 [illegible] 计算机工程学院 2023 级 计科 专业 3 班 学号：2340850112

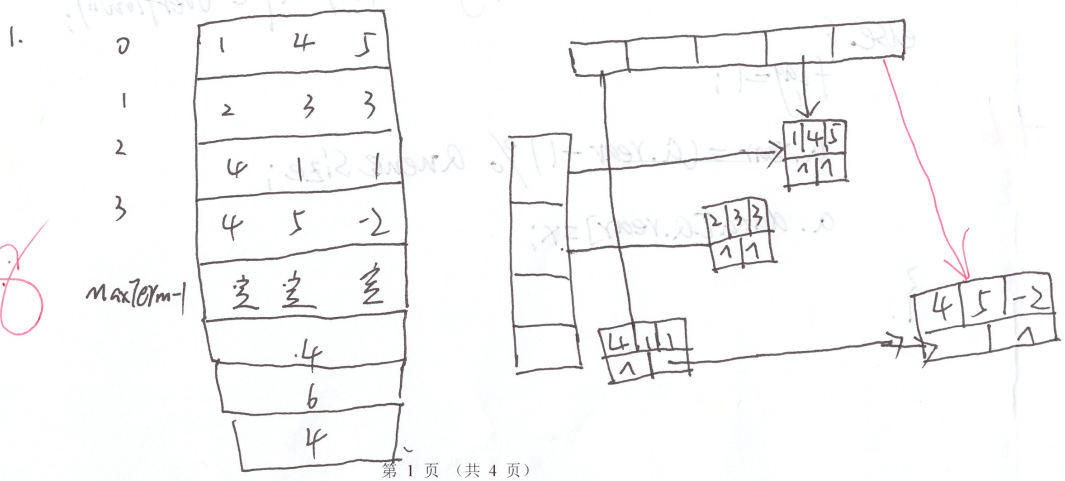
题号	一	二	三	总分
得分	26	30	19	75

得 分 26　一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	D	B	C	B	B	C	D	A
11	12	13	14	15					
A	C	D	A	D					

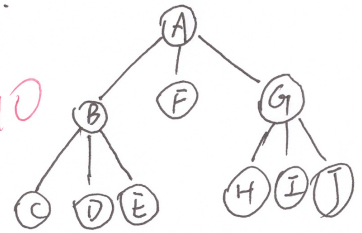
得 分 30　二、 结构分析题（每题 10 分，共 40 分）

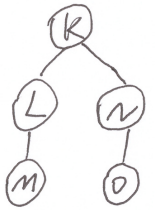
1.



+8

2.





+10

3.

1. 以 0, 01, 010 和 0101 为前缀的编码

2. 以 1, 00, 011, 0100 为前缀的编码.

+10

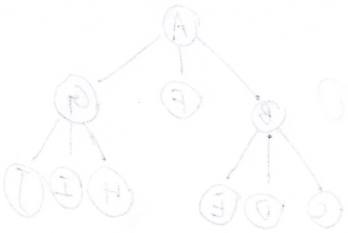
4. 设分支结点和叶子结点数分别是为 n_1 和 n_0.

∴ $n=n_0+n_1$, [illegible]

+2

2





得分	19

三、编程题（每题 10 分，共 30 分）

1.

```
typedef int ElemType;
typedef struct LNode{
ElemType data;
struct Londe *Link;
}*LinkList
int Search_K(LinkList list, int k){
LinkList p=list->link, q=list->link;
int count=0;
while(p!=NULL){
```

+3

2.

```
int Like(BiNode *s, BiNode *t)
{int same;
 if(s==NULL&&t==NULL)return 1;
else if((s==NULL&&t!=NULL)||(s!=NULL&&t==NULL))
return 0;
else{
   same=Like(s->lchild, t->lchild);
   if(same==1) same=Like(s->rchild, t->rchild);
   return same;
  }
}
```

+10

3.

```
void EnQueue(CirQueue &Q, DataType x) // flag为全局变量.
{ if(Q.front==Q.rear&&flag==1) printf("overflow");
else{
   flag=1;
   Q.rear=(Q.rear+1)%QueneSize;
   Q.data[Q.rear]=x;
}
```

+6

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚　审题人：王莉娜　试卷编号：第 A 套

姓名顾尚月　计算机工程学院　2023　级计算机科学与技术专业　3　班　学号：234085095

题号	一	二	三	总分
得分	24	28	16	68

得分 24

一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	D	B	BC	B	C	C	C	A
11	12	13	14	15					
AB	C	D	C	D					

得分 28

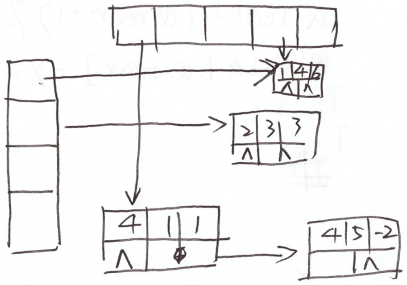
二、 结构分析题（每题 10 分，共 40 分）

1. (1) 三元组顺序表

0
1
2
3.
MaxTerm-1

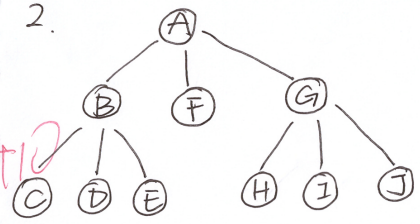
1	4	5
2	3	3
4	1	1
4	5	-2
4		
6		
4		

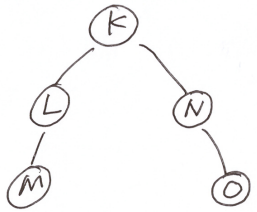
十字链表



+10

2.





+12

3. (1) 0, 01, 010, 0101

(2) 1, 00, 011, 0100, 1000

+8

4. $n = n_0 + n_k$

2. int like(Node *S, Node *t)

int Like(BNode *s, BNode *t)

{ int sa;

if (s==NULL && t=NULL)

return 1;

else if (s=NULL && t!=NULL) || (s!=NULL&& t==NULL))

return 0;

else {

sa=Like(s->lchild, t->lchild);

if (sa==1) sa=Like(s->rchild, t->rchild);

return sa;

}

}.

+10

3. void EQ(CQ &Q, P]x).

{ if (Q.front==Q.rear && flag==1)

printf("overflow");

else{ ~~flag~~

+6

flag=1;

Q.rear=(Q.rear+1)%QS;

Q.data[Q.rear]=x;

}

}.

~~#~~

得 分 | 16

三、编程题（每题 10 分，共 30 分）

1. ~~typedef int ElemType;~~

~~typedef struct LNode {~~

~~ElemType data;~~

~~Struct Lnode *Link;~~

}

江苏第二师范学院 2023—2024 学年第二学期

数据结构课程期中试卷

命题人：曹愚 审题人：王莉娜 试卷编号：第 A 套

姓名[illegible] 计算机工程学院 2023 级 计科 专业 1 班 学号：2340850001

题号	一	二	三	总分
得分	20	9	0	29

得 分 20

一、 选择题（每题 2 分，共 30 分）

1	2	3	4	5	6	7	8	9	10
B	B	D	B	C	C	C	C	B	A
11	12	13	14	15					
A	C	C	B	C					

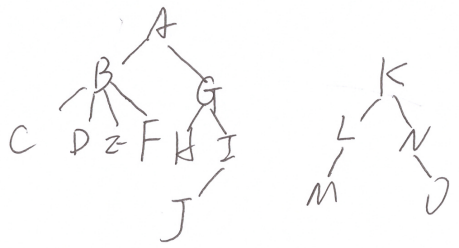
得 分 9

二、 结构分析题（每题 10 分，共 40 分）

1	4	5
2	3	3
4	1	1
4	5	-2

4
6
4

+3



+4

+0 (1) 0, 00, 010, 0101

(2) 1, 11, 100

得 分	

三、编程题（每题 10 分，共 30 分）

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 徐玮 计算机工程学院 2023 级 计科 专业 1 班 学号：2340850011

题号	一	二	三	四	总分
得分	20	10	48	20	98

得 分 20　一、单选题（每题 2 分，共 20 分）

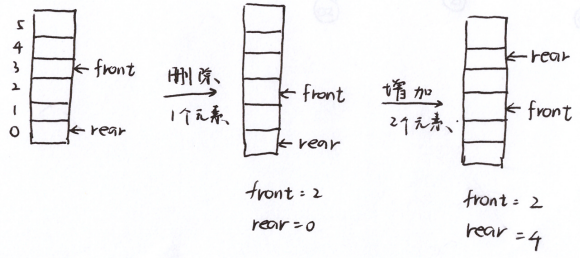
1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C

得 分 10　二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 36
5. {78, 19, 63, 28, 83, 84, 55, 69, 30, 89}

得 分 48　三、解答题（共 50 分）

1.



2. 顺序表：优点：方便查找元素位置
缺点：不方便进行插入和删除操作

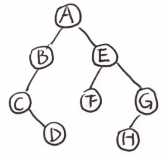
单链表：优点：方便进行插入和删除操作
缺点：不方便查找元素位置

4

(1) 顺序表存储

(2) 单链表存储

3.



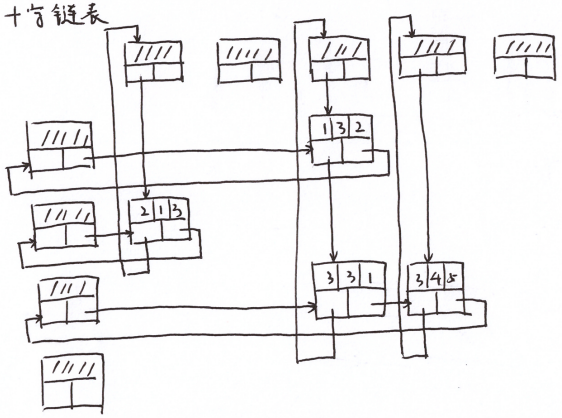
6

三元组顺序表

4.

序号	行号	列号	非零元素
1	1	3	2
2	2	1	3
3	3	3	1
4	3	4	5
	4（行数）		
	5（列数）		
	4（非零元素个数）		

十字链表



5.

j	0	1	2	3	4	5	6
next[j]	-1	0	1	0	1	2	3

匹配过程：

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14

a a b a b c a a b a a b c c b
a a b a a

第1次匹配，i=4，j=4 失败
i不动，j回溯到 next[4]，即 j=1

a a b a b c a a b a a b c c b
a a

第2次匹配，i=4，j=1 失败
i不动，j回溯到 next[1]，即 j=0

a a b a b c a a b a a b c c b
a

第3次匹配，i=4，j=0 失败
i=5，j回溯到 next[0]，即 j=-1

a a b a b c a a b a a b c c b
a

第4次匹配，i=5，j=0 失败
i=6，j回溯到 next[0]，即 j=-1

a a b a b c a a b a a b c c b
a a b a a b c

第5次匹配成功

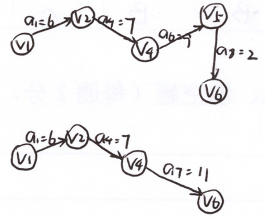
6. 事件最早和最迟开始时间　　活动最早和最迟开始时间

	Ve	Vl
V_1	0	0
V_2	6	6
V_3	4	5
V_4	13	13
V_5	22	22
V_6	24	24

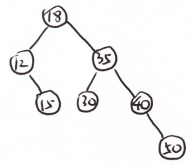
	e	l	
a_1	0	0	√
a_2	0	1	
a_3	4	5	
a_4	6	6	√
a_5	4	13	
a_6	13	13	√
a_7	13	13	√
a_8	22	22	√

当e=l时，活动为关键活动

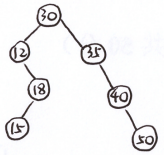
关键路径：



7.



或



8. 简单选择排序：

初始	4	1	3	5	2	7	6
第1趟	1	[4	3	5	2	7	6]
第2趟	1	2	[3	5	4	7	6]
第3趟	1	2	3	[5	4	7	6]
第4趟	1	2	3	4	[5	7	6]
第5趟	1	2	3	4	5	[7	6]
第6趟	1	2	3	4	5	6	7

快速排序：

初始	4	1	3	5	2	7	6
第1趟	2	1	3	4	5	7	6
第2趟	1	2	3	4	5	7	6
第3趟	1	2	3	4	5	6	7

8

得分	20

四、算法设计（每题 10 分，共 20 分）

1.
```
int ComBiTree ( BiNode * root )
{ BiNode * Q[100] , *p= NULL;
  int front=-1 , rear=-1 , flag=-1;
  if ( root== NULL) return 1;
  Q[++rear]= root;
  while ( front != rear)
  { p=Q[++front];
    if ( p-> lchild == NULL)
    { flag=0;
      if ( p-> rchild != NULL) return 0;
    }
    else
    { if (0==flag ) return 0;
      Q[++rear] =p-> lchild;
      if ( p-> rchild == NULL)
        flag=0;
      else Q[++front] = p-> rchild;
    }
    return 1;
  }
```


2.
```
int SimpleCircle ( MGraph *G , int indegree[] )
{ int count=0 , i, j , top=-1;
  for (i=0; i<G-> vertexNum ; i++)
  { if ( indegree[i]==0)
    { indegree[i]= top; top= i;}
  while (top!=-1)
  { count++; i= top;
    top= indegree[top];
    for (j=0; j< G-> vertexNum ; j++)
    { if ( G-> edge[i][j] !=1)
      { indegree[j] --;
        if ( indegree[j]==0)
        { indegree[j]= top;
          top=j;
        }
      }
    }
  if ( count < G-> vertexNum ) return 1;
  else return 0;
}
```

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 陈宗明　计算机工程学院 2023 级 计算机科学与技术 专业 1 班　学号：23408500

题号	一	二	三	四	总分
得分	20	8	49	20	97

得 分	20

一、单选题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C

得 分	8

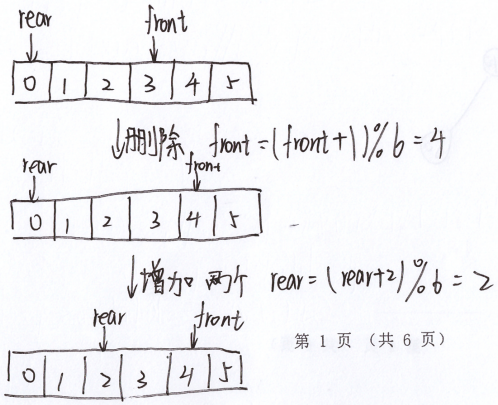
二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 36
5. 78 19 63 28 83 84 55 69 30 89

得 分	

三、解答题（共 50 分）

1.



6

2.

顺序表 优：随机读取性 已知下标读取任一元素时间复杂度为O(1)，对空间利用效率高数据紧凑

缺：在增删元素时要进行大量的元素移动时间复杂度为O(n)

可能造成对空间的浪费

构建时必须知道数据大小有接分配地址，且分配后不可扩展

单链表：优：在插入和删除元素时避免了大量元素的移动只改变逻辑位置

可扩展性

避免顺序表中空值对存储空间的浪费

缺：按位查找时间复杂度为O(n)

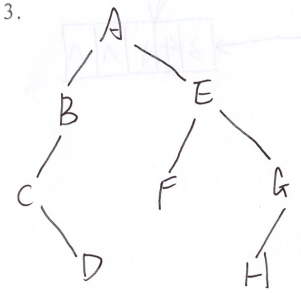
定义较顺序表复杂

6

(1) 顺序表

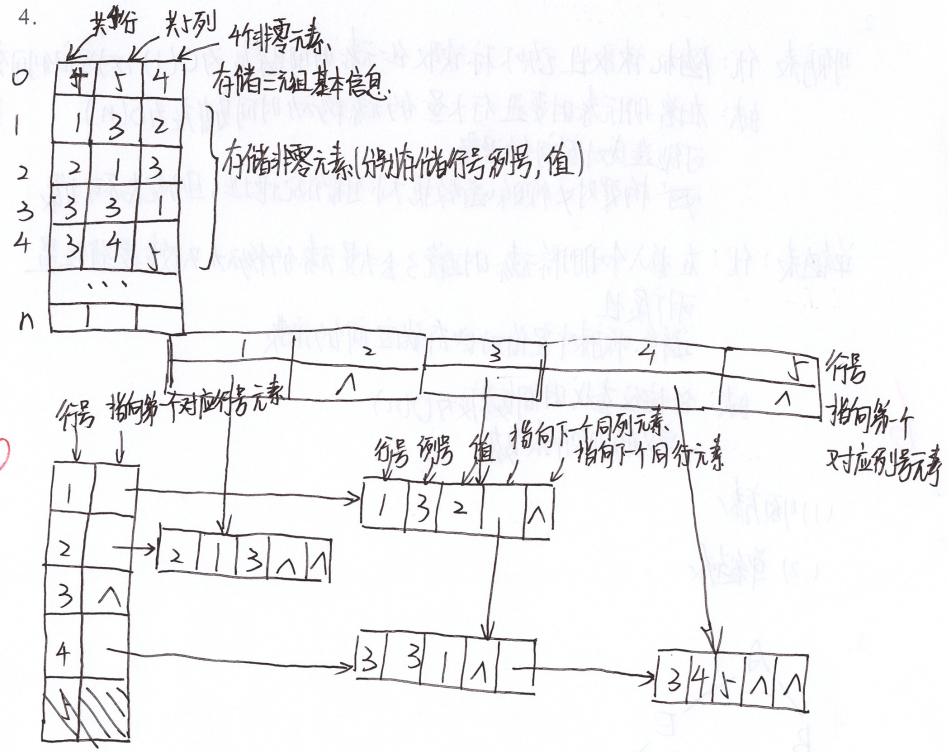
(2) 单链表

3.



6

4.



5.

aabaabc

next数组: -1 0 1 0 1 2 3

主串s: a a b a b c a a b a a b c c b

aab aa

aa

a

aabaabc

第 3 页 （共 6 页）

6.

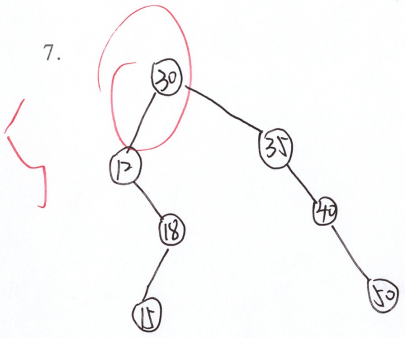
	最早	最迟
V_1	0	0
V_2	6	6
V_3	4	5
V_4	13	13
V_5	22	25
V_6	24	24

	最早	最迟
a_1	0	0
a_2	0	1
a_3	4	5
a_4	6	6
a_5	4	13
a_6	13	13
a_7	13	13
a_8	22	22

关键路径 $V_1 \to V_2 \to V_4 \to V_5 \to V_6$

$V_1 \to V_2 \to V_4 \to V_6$

7.



第 4 页 （共 6 页）

8.

简单选择 4 1 3 5 2 7 6
1 4 3 5 2 7 6
1 2 3 5 4 7 6
1 2 3 5 4 7 6
1 2 3 4 5 7 6
1 2 3 4 5 7 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7

快速排序 4 1 3 5 2 7 6
2 1 3 4 5 7 6
~~1 2 3 4 5 7 6~~
1 2 3 4 5 7 6
1 2 3 4 5 6 7

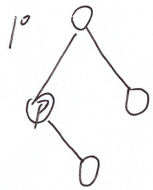
8

得 分	20

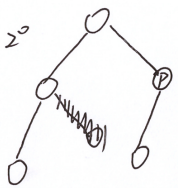
四、算法设计（每题 10 分，共 20 分）

1.
```
int Tree(node* root){
node* G[100], *p;   //初始化队列用于层序遍历
int front=0, rear=0, flag=0 //初始化队列指针 和标记符
if(root==NULL)
return 1; //空树为完全二叉树
G[rear++] = root;   //根入队
while(front<rear){ //当队不空时
 p= G[front++]; //队头出列 判断
if(p->lchild == NULL){
   if(p->rchild != NULL)
      return 0; //1°当p左孩子为空但有右孩子时
}
else{
   if(flag==1)
   return 0; // 2°当p左踢的右孩子为空但p有左孩子时
   G[rear++] = p->lchild;}
if(p->rchild == NULL)
flag==1; //当前元素右孩子为空 作标记 用于右踢的判断
else
G[rear++] = p->rchild;}      return 1;}
```

1°



2°



两种非完全二叉树

2. 使用拓扑排序 若拓扑排序中元素 小总元素个数则存在回路

```
int simpleCircle(MGraph*G, int indegree){
int i;
buildstack(s); //初始化栈用于存储入度为0的元素
for(i=0; i<G->Num; i++)
 if(indegree[i]==0)
 push(s,i); // 把初始时入度为0的元素压入栈
int count = 0; //用于计数
while(!isEmpty(s)){ //当栈非空时
i=pop(s); //把栈顶取出 并把数值存入i中
int j;
for(j=0; j<G->Num; j++)
 if(G->edge[i][j]==1) //去掉以i为边尾的边
   if(--indegree[j]==0)
    push(s,j); //压入以新型的入度为零的元素
count ++; //成功排一次 计数加1
}
if(count<G->Num)
return 0; //完成拓扑排序元素个数 小于总元素个数，说明存在环
else
return 1;
}
```

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 王天宇 计算机工程学院 2023 级 计算机科学与技术 3 班 学号：2340850091

题号	一	二	三	四	总分
得分	20	10	46	8	84

得分 20　一、单选题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C

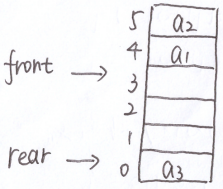
得分 10　二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 36
5. {78, 19, 63, 28, 83, 84, 55, 69, 30, 893}

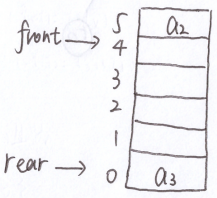
得分 46　三、解答题（共 50 分）

1. 解：

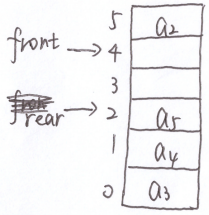
初始状态：



删除一个元素后：



再增加两个元素后：



由操作示意图知

rear=2，front=4

+6

2. 解：顺序表 ①优点：不需要额外为数据元素之间的逻辑关系增加额外存储空间，存储密度大，可实现数据的随机存取

②缺点：插入与删除数据元素时需要移动大量数据元素。

顺序表处理问题时易造成存储空间"碎片化"

单链表 ①优点：插入与删除操作时仅需要指针的变化，不需要移动数据元素，适用于不知道表长的情况

单链表是动态存储方式

②缺点：需要指针的结构性"开销"，存取数据时只能顺序存取。

(1) 用顺序表存取更好，因为表长度稳定且很少进行插入与删除操作，读取元素快

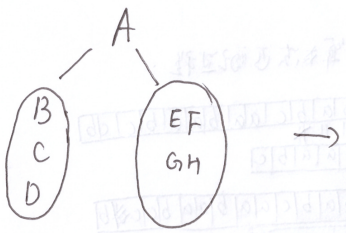
(2) 用单链表存取更好，因为 n 个线性表并存，且各表长度动态变化。

+6

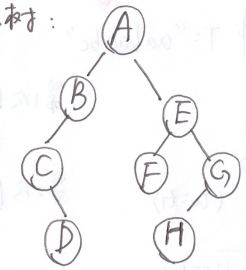
3.

解：前序遍历：ABCDEFGH

中序遍历：CDBAFEHG



二叉树：



+6

4. 解：

① 三元组顺序表：

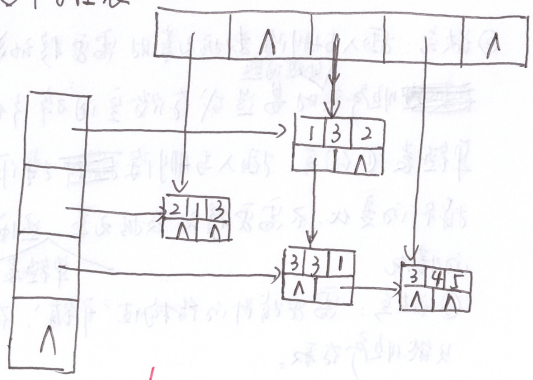
下标	行数	列数	非零元素
0	1	3	2
1	2	1	3
2	3	3	5
3	3	4	5

行数 4

列数 5

非零元素个数 4

② 十字链表：



+6

5. 解：模式串 T="aabaabc"

next 数组：

j=0 next[j]=-1

j=1 next[j]=0

j=2 next[j]=1 ($t_0=t_1$)

j=3 next[j]=0

j=4 next[j]=1 ($t_0=t_3$)

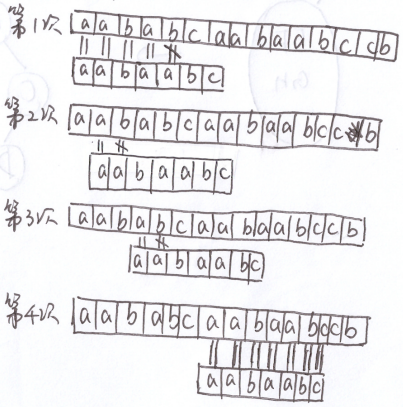
j=5 next[j]=2 ($t_0t_1=t_3t_4$)

j=6 next[j]=3 ($t_0t_1t_2=t_3t_4t_5$)

故该模式下next数组为

{-1,0,1,0,1,2,3}

KMP算法匹配过程：



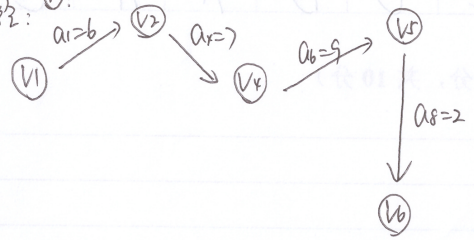
+6

6. 解：

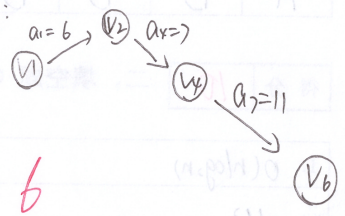
活动	v_1	v_2	v_3	v_4	v_5	v_6
最早开始时间	0	6	4	13	22	24
最迟开始时间	0	6	5	13	22	24

活动	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
最早开始时间	0	0	4	6	4	13	13	22
最迟开始时间	0	1	5	6	13	13	13	22

关键路径：①：



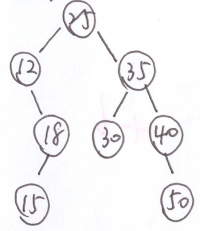
②：



+6

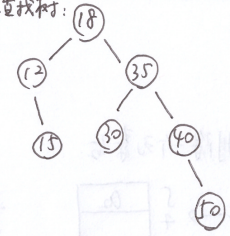
7. 解：在根结点为18的左子树中，18代替被删除根结点为35

原二叉查找树：



删除元素35

新二叉查找树：



+6

8. 解：数据序列{4,1,3,5,2,7,6}

①简单选择排序

初始状态 4,1,3,5,2,7,6

第1次：1435276

第2次：1345276

第3次：1245376

第4次：1235476

第5次：1234576

第6次：1234567

②快速排序：

初始状态 4,1,3,5,2,7,6

选取最左边元素4作为轴值 ↑ 4,1,3,5,2,7,6

第1次：[2,1,3],4[5,7,6]

第2次：1234[5,7,6]

第3次：1234567

+4

得 分	8

四、算法设计（每题 10 分，共 20 分）

1. 解：

```
BiNode ~~*~~ ~~int~~ *Node(int i,j,root, newLptr[], newRptr[])
~~if root~~ int i,j,root;
  { if root=Null;
      return; }
for(i=0; i<n; i++)
  for(j=0; j<n, j++)
if(root->lchild=0) {rchild=0;}
if(root->lchild==0 && root->rchild==0)
  { ~~rchild~~ root->lchild->lchild==0 && root->lchild->rchild==0 }
```

+4

2.

```
int SimpleCircle (MGraph *G, int indegree[])
{ int i,j;
  for(i=0; i<n; i++)
  int indegree[100];
  for(n=0; n≤100; n++)
{
  if( i!=0&& j!=0 )
      return 1; //有回路
  else return 0; //无回路
}
```

+4

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平 审题人：王莉娜 试卷编号：第 B 套

姓名 [illegible] 计算机工程学院 2023 级 计算机科学与技术 专业 3 班 学号：2360800098

题号	一	二	三	四	总分
得分	20	10	44	6	80

得 分 20 一、单选题（每题 2 分，共 20 分）

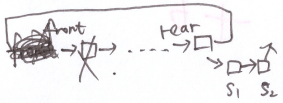
1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C

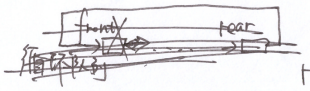
得 分 10 二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 36
5. 78, 19, 63, 28, 83, 84, 55, 69, 30, 89

得 分 44 三、解答题（共 50 分）

1.





rear 的值为 2，front 的值为 4.

+6

2. 顺序表：优点：可随机存取，存储密度大
缺点：不易进行插入和删除操作
单链表：优点：插入和删除操作时仅需移动指针，动态存储
缺点：不能随机存取

(1) 顺序存储结构更好，因为线性表很少进行插入和删除操作，顺序存储结构便于随机存取，

(2) 链式存储结构更好，因为线性表的长度动态发生变化，链式存储结构更适合动态存取。

+6

3.

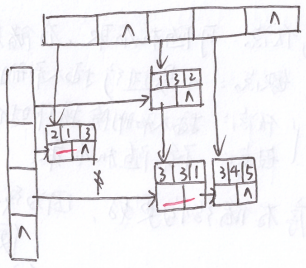


+6.

4. 三元组顺序表

	行	列	非零元素
1	1	3	2
2	2	1	3
3	3	3	1
3	3	4	5
行	#4		
列	#5		
非零个数	4		

十字链表



+5

5. $next[j]=\begin{cases}-1 & j=0 \\ \max\{k \mid 1\le k<j \text{ 且 } T[0]\dots T[k-1]=T[j-k]\dots T[j-1]\} & \text{集合非空} \\ 0 & \text{其他情况}\end{cases}$

next[j]={-1,0,1,0,1,0,0,1,2,3,4,5,6,0,0}

i=0 j=0 a a b a b c a a b a a b c c b

第1趟 i=1 j=1 a a b a a b c

回溯

匹配失败 a a b a b c a a b a a b c c b

第2趟 i=1 j=0 a a b a a b c

回溯

匹配失败 a a b a b c a a b a a b c c b

第3趟 i=3 j=0 a a b a a b c

回溯

匹配失败 a a b a b c a a b a a b c c b

第4趟 i=6, j=3 a a b a a b c

匹配成功.

+3

6.

事件	V_1	V_2	V_3	V_4	V_5	V_6
最早开始时间	0	6	4	13	24	24
最迟开始时间	0	6	5	13	22	24

活动	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
最早开始时间	0	0	4	6	4	13	13	22
最迟开始时间	0	1	5	6	13	13	13	22

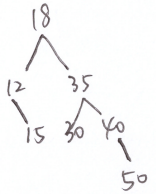
关键活动为 a_1, a_4, a_6, a_7, a_8

A0E网的所有关键路径为 ① $V_1\to V_2\to V_4\to V_5\to V_6$

② $V_1\to V_2\to V_4\to V_6$

+6

7.



+6

8. ① 简单选择排序 第一趟：1, 4, 3, 5, 2, 7, 6
第二趟：1, 2, 3, 5, 4, 7, 6
第三趟：1, 2, 3, 4, 5, 7, 6
第四趟：1, 2, 3, 4, 5, 6, 7

② 快速排序 第一趟：2, 1, 3, 4, 5, 7, 6.
第二趟：1, 2, 3, 4, 5, 7, 6
第三趟：1, 2, 3, 4, 5, 6, 7

+6

得 分	6

四、算法设计（每题 10 分，共 20 分）

1.
```
void BFS (BiNode List *p, p)
{ if ( L!==NULL)
    if (p->lchid == NULL && p->rchild!==NULL)   // 该结点无左孩子，但有右孩子，一定不是完全二叉树
        return -1;
    if (p->rchild==NULL) && p->next!==NULL)   // 该结点无右孩子，但后继结点有孩子，一定不是完全二叉树
        return -1;
    else
        return 0;
}
```

+3.

2.
```
int SimpleCircle (MGraph *G, int degree[])
{ int indegree[n];
  ~~if (V[i][j] == NULL)~~
  ~~return i++;~~
  for (i=0; i<=n; i++)
    for (j=0; j<=n; j++)
      if (V[i][j] == NULL)
        ~~print~~ return 1;
    else
      return 0;
}
```

//遍历数组.
//判断路径是否存在.
//拓扑排序没有回路

+3

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 金佳雯 计算机工程学院 2023 级 计算机科学与技术 专业 3 班　学号：2340850083

题号	一	二	三	四	总分
得分	20	6	38	14	78

得 分	20

一、单选题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C.

得 分	6

二、填空题（每题 2 分，共 10 分）

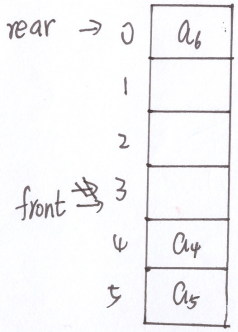
1. $n\log_2 n$
2. 112.
3. 栈
4. 36.
5. {84, 19, 63, 28, 83, 78, 55, 69, 30, 89}.

得 分	38

三、解答题（共 50 分）

+6

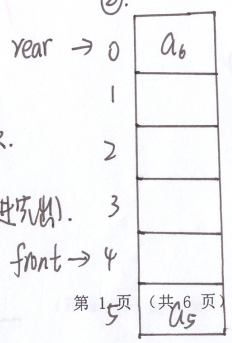
1. ①.开始



删除元素.

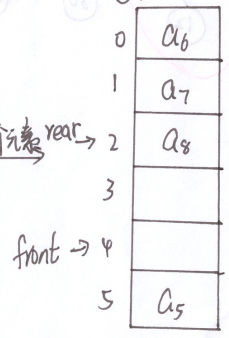
(队列先进先出).

②.



增加2个元素

③.



2. 顺序表：

优点：①. 随机存取，能够随机存取任意元素.

②. 不用存储元素之间的逻辑关系.

缺点：①. 增加或删除元素时要移动大量元素.

②. ~~不能~~ 由于顺序表需要连续空间进行存储，可能会导致内存"碎片"化.

单链表：

优点：①. 增加或删除元素时，不必移动元素，只需修改指针即可.

②. 动态存储，不必事先知道线性表长度.

缺点：①. 单链表存储需要指针，会造成结构性开销.

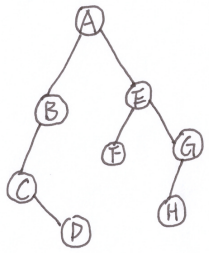
②. ~~或存取~~ 按位查找时，只能通过遍历.

(1). 使用顺序表，因为已知线性表长度，~~或~~。单链表查找元素时比顺序表慢。

(2). 使用单链表，因为各表长度会发生变化，使用顺序表可能存不下.

+6

3.

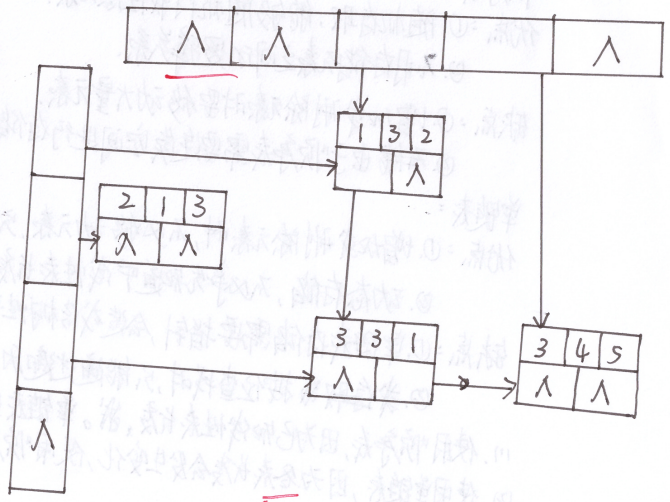


+6

4. 三元组.

序号	行号	列号	非零元素
0	1	3	2.
1	2	1	3
2	3	3	1
3	3	4	5
行数.	4		
列数.	5		
非零个数	4.		

十字链表.



+5

5. next数组{-1, 0, 1, 0, 1, 2, 3}.

① a a b a b c a a b a a b c c b.
a a b a a

② a a b a b c a a b a a b c c b.
a a

③ a a b a b c a a b a a b c c b.
a a

④ a a b a b c a a b a a b c c b.
a a b a a b c

+3

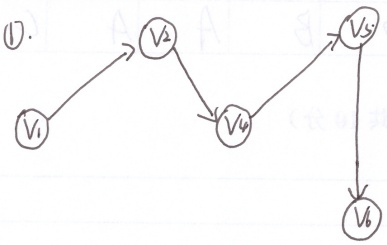
6.

	V_1	V_2	V_3	V_4	V_5	V_6
最早开始	0	6	4	13	22	24
最迟开始	0	6	5	13	22	24

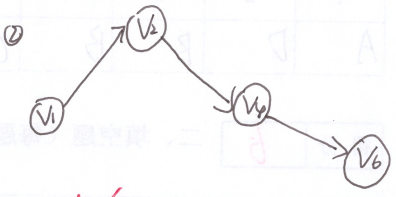
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
最早开始	0	0	4	6	4	13.	13.	22
最迟开始	0	1	5	6	5	13	13	22.

关键路径：

①.

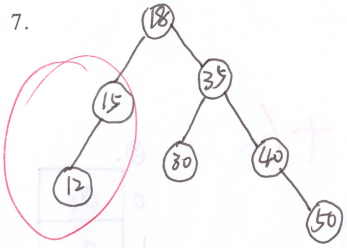


②



+6

7.



+4

8. 1. 简单选择排序

第一趟：1, 4, 3, 5, 2, 7, 6.
第2趟：1, 2, 4, 3, 5, 7, 6.
第3趟：1, 2, 3, 4, 5, 7, 6.
第4趟：1, 2, 3, 4, 5, 7, 6.
第5趟：1, 2, 3, 4, 5, 7, 6
第6趟：1, 2, 3, 4, 5, 6, 7.

2. 快速排序

第1趟：1, 3, 2, 4, 5, 7, 6.
第2趟：1, 3, 2, 4, 5, 7, 6.
第3趟：1, 2, 3, 4, 5, 6, 7.

+2

得分 14

四、算法设计（每题 10 分，共 20 分）

1.
```
int ComBiTree(BiNode *root).
{
    BiNode *Q[100], *p=NULL;
    int rear=-1, front=-1, flag=1;
    if(root==NULL) return 1;
    Q[++rear]=root;
    while(rear!=front).
    {
        p=Q[++front];
        if(p->lchild==NULL).
        {
            flag=0;
            if(p->rchild!=NULL). return 0;
        }.
        if(0==flag) return 0;
        Q[++rear]=p->lchild;
        if(p->rchild==NULL) flag=0;
        else Q[++front]=p->rchild;
    }.
```

+8

2.
```
int SimpleCircle(MGraph *G, int indegree[]).
{
    int i, j, top=-1;
    BiNode *p=NULL;
    for(i=0; i<G->vertexNum; i++).
    {
        if(indegree[i]==0)
        {
            indegree[i]=top;
            top=i;
        }.
    }.
    while(top!=-1).
    {
        for(j=0; j<G->vertexNum; j++).
        {
            if(G->edge[i][j]==0). return 0;
            if(indegree[j]==0)
            { indegree[j]=top; top=j; }
```

+6

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 高浩 计算机工程学院 2023 级 计科 专业 3 班 学号：2340350092.

题号	一	二	三	四	总分
得分	16	10	35	9	70

得 分 16　一、单选题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	C	A.	A.	D.

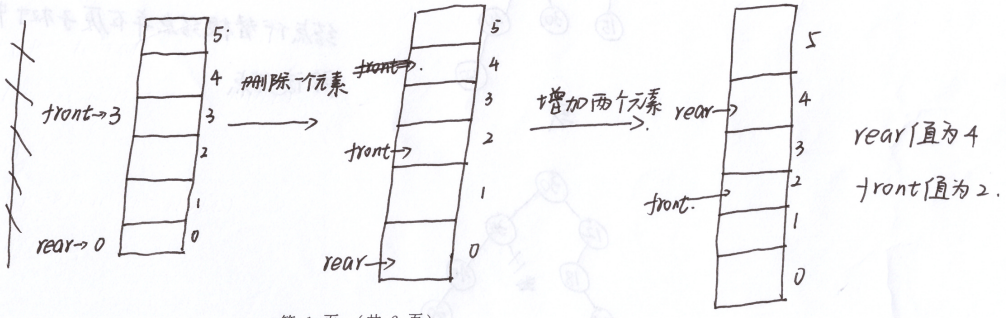
得 分 10　二、填空题（每题 2 分，共 10 分）

1.	$O(n\log_2 n)$
2.	112
3.	栈
4.	36
5.	78 19 63 28 83 84 55 69 30 89

得 分 35　三、解答题（共 50 分）

+6

1.



2. 顺序表优点：存取方便

缺点：删除、插入等操作需移动大量元素

需提前预留充足空间

单链表优点：删除、插入等操作方便，不必提前预留空间

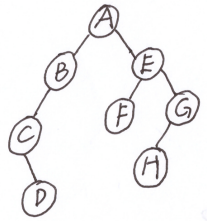
缺点：读取元素不方便.

(1). 顺序表存储

+6

(2). 单链表存储.

3.



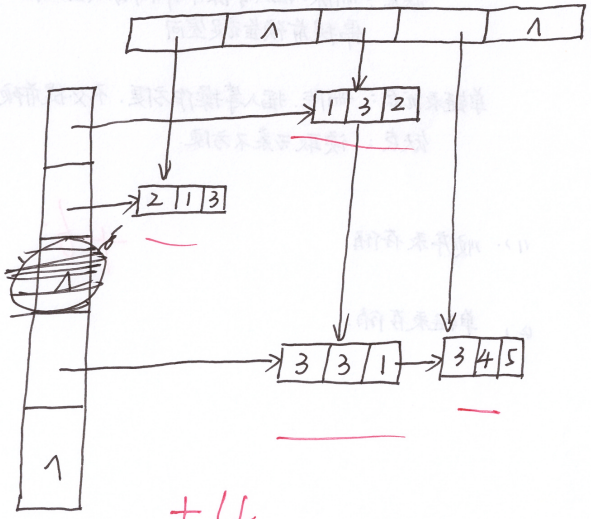
+6

4.

三元顺序表.

行	列	元素
1	3	2
2	1	3
3	3	1
3	4	5
4（行数）		
5（列数）		
4（元非零元素）		

十字链表.



+4

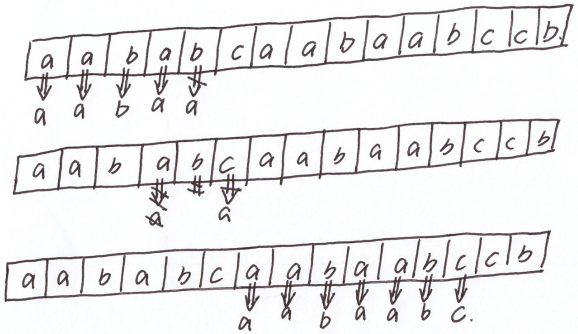
十字链表



+2

5.

next数组：-1 0 1 0 1 0 0 1 2 3 4 2 0 0 0.

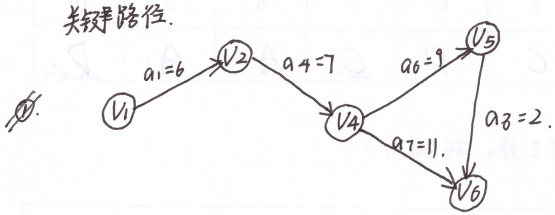


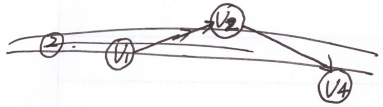
6.

	最早开始时间	最晚开始时间
V_1	0	0
V_2	6	6
V_3	4	5
V_4	13	13
V_5	22	22
V_6	24	24

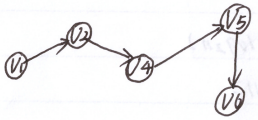
+4

关键路径.

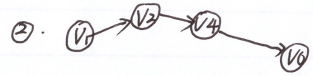




①.

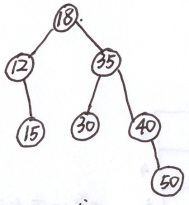


②.

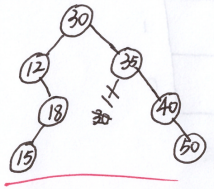


7.

+5



或



用左子树中最大结点或右子树中最小结点代替根结点并在原子树中删除该结点.

8.

简单选择排序.

4 1 3 5 2 7 6.

1 4 3 5 2 7 6.

1 3 4 5 2 7 6.

1 2 4 5 3 7 6.

1 2 3 5 4 7 6.

1 2 3 4 5 7 6.

1 2 3 4 5 6 7.

+2

快速排序.

4 1 3 5 2 7 6.

1 4 3 5 2 7 6.

~~1 4 3 5 2 7 6.~~

1 2 3 5 4 7 6.

1 2 3 4 5 7 6.

1 2 3 4 5 6 7.

得分	9

四、算法设计（每题 10 分，共 20 分）

1.

```
int comBiTree(BiNode *root).
{ BiNode *Q[100], *p=NULL;
  int front=-1, rear=-1, flag=1.
  if(root==NULL)
  ~~printf("%2c~~
  return 1;
  Q[++rear]=root;
  while(front!=rear)
  {
    p=Q[++front];
    ~~if(p->lchild~~
    if(p->lchild ~~==0~~ ==NULL)
  { flag=0
    if(p->rchild!=NULL)
    return 0;
    else{
    if(flag==0) return 0;
    Q[++rear]=p->lchild;
    if(p->rchild==NULL)
    flag=0.
  }
    Q[++rear]=p->rchild;
  }
}.
```

+8

2.

~~void simplecycle(MGraphy. AL, interger[]).~~

~~void~~ int SimpleCircle(MGraph *G, int indegree[]).

```
{
  int i, j;
  ~~Node *p=NULL~~
```

+1

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 [illegible] 计算机工程学院 2023 级 计科 专业 3 班 学号：2340850111

题号	一	二	三	四	总分
得分	14	8	34	9	65

得 分 14　一、单选题（每题 2 分，共 20 分）

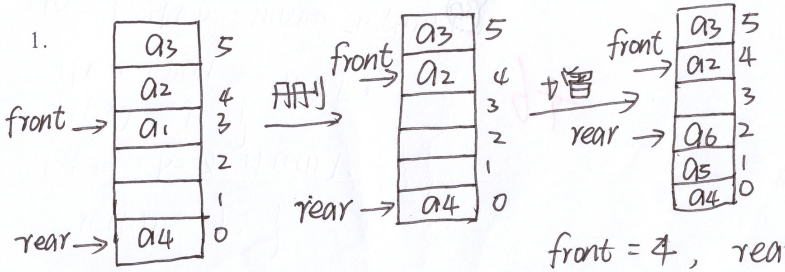
1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	C	D	A	B

得 分 8　二、填空题（每题 2 分，共 10 分）

1. O(nlog2n)
2. 112
3. 栈
4. 36
5. 78，19，63，30，84，89，55，69，28，83

得 分 　　三、解答题（共 50 分）

1.



+6

2. ①顺序表可以快速找到所需元素位置，有限自[illegible]存取的特性，但是所占空间必须是连续的，在操作时，空间过[illegible]大元素过少会造成内存浪费。

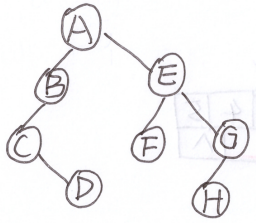
②单链表的空间内存可以是不连续的，可以节省内存的使用，但不能快速查找所需元素。

(1). 顺序表

(2). 单链表.

+6

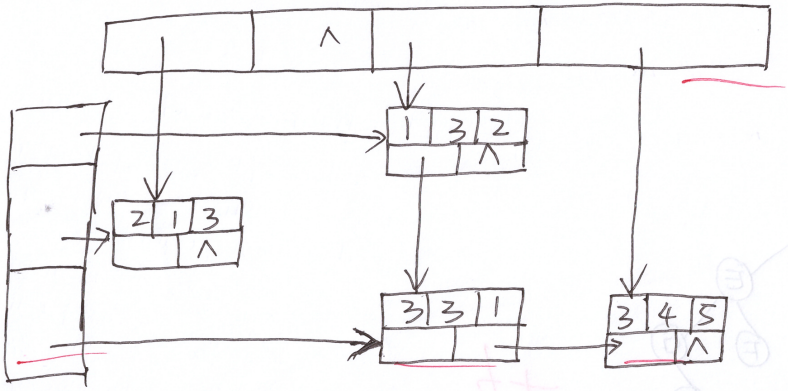
3.



+6

4.

下标	行	列	非零元素
0	1	3	2
1	2	1	3
2	3	3	1
3	3	4	5
行数 4			
列数 5			
非零元素 4			



5.

+5

+0.

6. 最早 V_1, V_3, V_6 $T=15$.

最迟 $T=23$

① $V_1\ V_2\ V_4\ V_5\ V_6$

② $V_1\ V_2\ V_4\ V_6$

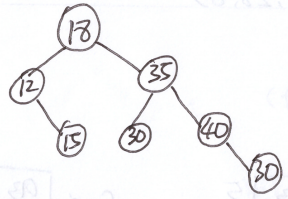
③ $V_1\ V_3\ V_4\ V_6$

④ $V_1\ V_3\ V_6$

⑤ $V_1\ V_3\ V_4\ V_5\ V_6$.

+4

7.



+6

8.

简单排序

① 1 4 35276
② 1 2 3 5 4 76
③ 1 2 3 4 5 76
④ 1 2 3 4 5 76
⑤ 1 2 3 4 5 67

快速排序

① 4 1 4 3 5 2 7 6
② 1 3 4 5 2 7 6
③ 1 3 4 2 5 6 7
④ 1 3 2 4 5 6 7
⑤ 1 2 3 4 5 6 7

+3

得分 9

四、算法设计（每题 10 分，共 20 分）

1.
```
int ComBiTree ( TNode *root) {
TNode * Q[100], *p=NULL;
int rear=-1, front=-1, flag=1;
if( root==NULL) return 1;
Q[++rear] = root;
while (front != rear) {
p= Q[++front];
if(p->lchild == NULL){
flag=0;
if( p->rchild != NULL)
return 0; }
else { if(p->rchild==NULL) return 0;
if( p->rchild == NULL)
flag = 1; }
Q[++rear] = p->rchild; } }
return 1; }
```

+6

2.
```
int SimpleCircle (MGraph *G, int indegree[]){
{ int i;
for(i=0; i<G->vexnum; i++)
{ if(indegree[i]==0)
if( indegree[i] ==0)
int SimpleCircle (MGraph *G, int indegree[])
{ int i;
for (i=0; i<G->vexnum, i++)
{ if (indegree[i] == 0)
return 0; }
for ( i=0; i<G->vexnum, i++)
{
```

+3

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 陈紫萱 计算机工程学院 2023 级 计科 专业 5 班 学号：2340850107

题号	一	二	三	四	总分
得分	20	8	29	2	60

得 分 20　一、单选题（每题 2 分，共 20 分）

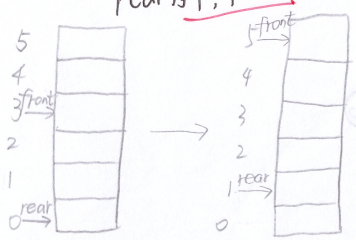
1	2	3	4	5	6	7	8	9	10
A	D	B	B	C	D	B	A	A	C

得 分 8　二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 244
5. 78 19 63 28 83 84 55 69 30 89

得 分 29　三、解答题（共 50 分）

1. rear为1，front为5

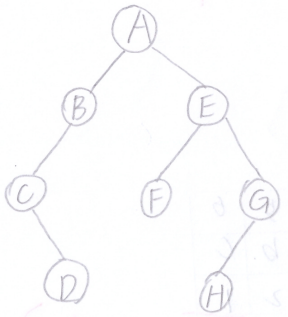


2. 顺序表：优点：①元素的物理地址与逻辑地址保持一致 ②能快速的查找元素 ③不需要额外开辟存储指针的空间
缺点：插入、删除元素不方便
单链表：优点：①表的长度可发生变化，无需元素逐个移动
缺点：需额外开辟空间存储指针

(1) 采取顺序表结构更好
(2) 采取单链表结构更好

+6

3.



+6

4.

	1	2	3	4	5
1	0	0	2	0	0
2	3	0	0	0	0
3	0	0	1	5	0
4	0	0	0	0	0

行号	列号	非零元
1	3	2
2	1	3
3	3	1
3	4	5
4（行数）		
5（列数）		
4（非零元个数）		

+3.

5.

j	0	1	2	3	4	5	6
模式	a	a	b	a	a	b	c
next	-1	0	1	0	1	2	1

+3

① a a b a b c a a b a a b c c b
  ‖ ‖ ‖ ✗ ✗ ✗
  a a b a a b c

② a a b a b c a a b a a b c c b
  ‖ ‖ ‖ ‖ ‖ ‖
  a a b a a b c

6.

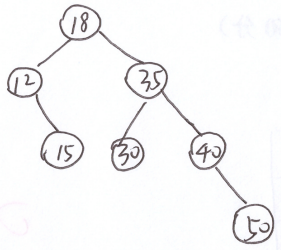
	V_1	V_2	V_3	V_4	V_5	V_6
Ve[k]	0	6	4	13	22	24
Vl[k]	0	6	5	13	22	24

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
e[i]	0	0	4	6	4	13	13	22
el	0	1	5	6	13	13	13	22

+3

该AOE网的所有关键路径是 a_1, a_4, a_6, a_7, a_8

7.



+6

8. 简单选择排序

[4] 1 3 5 2 7 6
第一趟 [1 4] 3 5 2 7 6
第二趟 [1 2 4] 3 5 7 6
第三趟 [1 2 3 4] 5 7 6
第四趟 [1 2 3 4 5] 7 6
第五趟 [1 2 3 4 5 6] 7

快速排序

4 1 3 5 2 7 6
第一趟 2 1 3 5 4 7 6
第二趟 1 2 3 5 4 7 6
第三趟 1 2 3 4 5 7 6
第四趟 1 2 3 4 5 6 7

+2.

得分	27

四、算法设计（每题 10 分，共 20 分）

1.

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
Typedef DataType ;
Typedef struct BiNode
{     DataType data;
      struct BiNode*(lchild,rchild);
} BiNode
Void PreNode *( lchild, rchild)
{ if(lchild==NULL)  return;
```

+2

2.

```
#include <stdio.h>
#include <stdlib.h>
  int SimpleCircle (MGraph *G, int indegree[])
  int n;
```

+1

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名：杨君婷　计算机工程学院　2023 级　计算机 专业　3 班　学号：2340850113

题号	一	二	三	四	总分
得分	18	8	22	8	56

得分 18　一、单选题（每题 2 分，共 20 分）

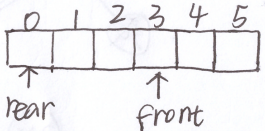
1	2	3	4	5	6	7	8	9	10
A	D	B	C	C	D	B	A	A	C

得分 8　二、填空题（每题 2 分，共 10 分）

1. $O(n\log_2 n)$
2. 112
3. 栈
4. 36
5. {30, 19, 63, 78, 89, 84, 55, 69, 28, 83}

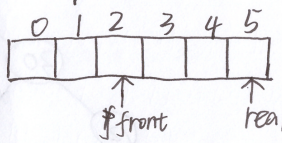
得分 22　三、解答题（共 50 分）

1. 解：原队列为

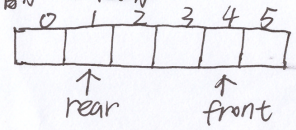


+4

删除一个元素：



增加两个元素：



由操作示意图可知从队列中删除一个元素再增加两个元素后，rear 的值为 1，front 值为 4。

2. 答：(1)：顺序表优点：是静态分配，有一定的存储空间，

缺点：消耗时间太长。

所以线性表的当长度基本稳定，少进行插入和删除时，但要求以最快速度按位置存取线性表中的元素时，采用顺序表来存储更好些。

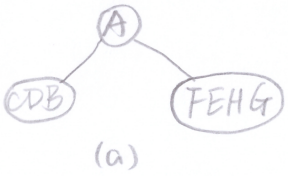
(2)：单链表：优点：能动态开辟存储空间。

缺点：

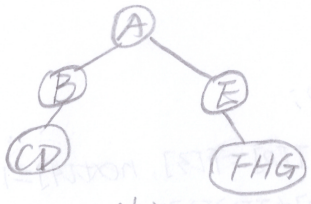
+4

如果 n 个线性表同时并存，且在处理过程中各表长度会动态发生变化时，采用单链表来存储更好些。

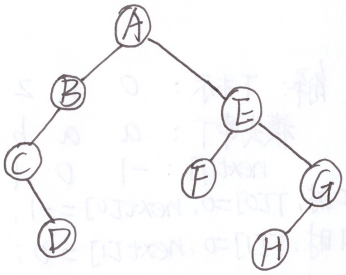
3. 解：根据前序遍历可知根结点为 A，由中序遍历可知根结点的左子树和右子树分别为 B、E，同理其余结点位置也可推出，因此二叉树如下图(c)所示：（图 a、b 为推理过程）。



(a)



(b)



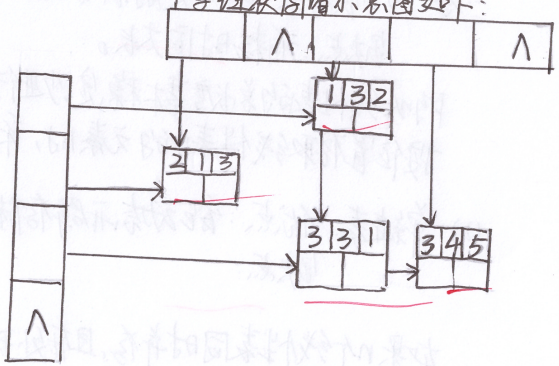
(c)

+6

4. 解：三元组顺序表如下图：

下标	行	列	非零元素
0	1	3	2
1	2	1	3
2	3	3	1
3	3	4	5
4(行数)			
5(列数)			
4(非零元素个数)			

十字链表存储示意图如下：



+5

5. 解：

下标	0	1	2	3	4	5	6
模式串T	a	a	b	a	a	b	c
next[j]	-1	0	1	0	1	2	

①：j=0时，T[0]=0，next[0]=-1；

②：j=1时，T[1]=0，next[1]=0；

③：j=2时，T[0]=T[1]，next[2]=1；

④：j=3时，T[0][1]≠T[1][2]，T[0]≠T[2]，next[3]=0；

⑤：j=4时，T[0][1][2]≠T[1][2][3]，T[0][1]≠T[1][2]，T[0]=T[3]，next[4]=1；

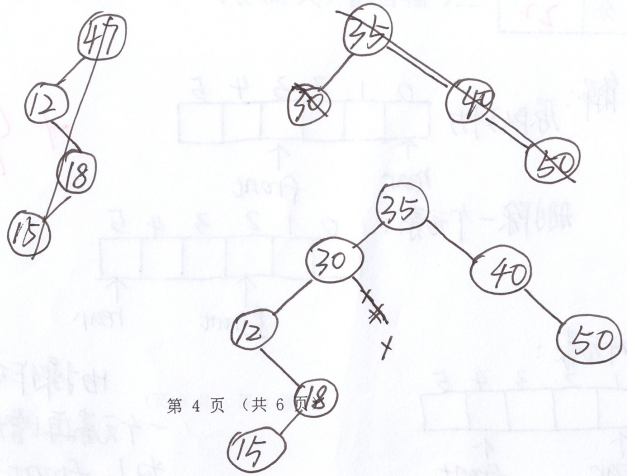
⑥：j=5时，T[0][1][2][3]≠T[1][2][3][4]，T[0][1][2]≠T[2][3][4]，
T[0][1]=T[3][4]，next[5]=2；

⑦

+3

6. 答：关键路径：{v_1, v_2, v_3, v_4, v_5, v_6}

7. 解：删除元素25后的二叉查找树为：



8. 解：(1,4,3,5,2,7,6)
(1,2,3,5,4,7,6)
(1,2,3,5,4,7,6)
(1,2,3,4,5,7,6)
(1,2,3,4,5,6,7)

得分 8

四、算法设计（每题 10 分，共 20 分）

1. ~~void BiTree(Graph *A, Graph~~

```
void BiTree(Graph *A, int n)
{ int i,j;
  Node *p=NULL;
```

+2

2. 解：

~~void ListMat~~

```
void ListMat(MGraph *A,AL Graph *B)
{ int i,j;
  Node *p=NULL;
 A->vertexNum=B->vertexNum;
 A->edgeNum=B->edgeNum;
 for(i=0;i<A->vertexNum;i++)
 for(j=0;j<A->vertexNum,j++)
   ~~A->edge[i][j]~~
   A->edge[i][j]=0;
 for(i=0;i<A->vertexNum;i++)
 { A->vertex[i]=B->adjlist[ij.vertex;
   ~~p=B->v~~
   p=B->adjvex;
  while(p!=NULL)
   {
     A->edge[i][j]=1;
   p=p->next;
   }
   }
   }
```

+5

江苏第二师范学院 2023—2024 学年第二学期

数据结构 课程期末试卷答题纸

命题人：陶雯、吴利平　审题人：王莉娜　试卷编号：第 B 套

姓名 杨武 计算机工程学院 2023 级 计科 专业 3 班 学号：2340850112

题号	一	二	三	四	总分
得分	8	6	13	0	27

得 分 8　一、单选题（每题 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
A	D	B	A	C	B	D	D	B	A

得 分 6　二、填空题（每题 2 分，共 10 分）

1. O(n log₂n)
2. 112
3. 双链表
4. 12
5. 63

得 分 13　三、解答题（共 50 分）

1.

0	1	2	3	4	5	
rear			front			

删除一个后 rear 值 0→1

增加一个后 front 值 3→5

0	1	2	3	4	5
	rear				front

2.

顺序表：优：总长度确定，方便快速查找，存取线性表中元素，

缺：长度固定可能浪费存储空间，也可能出现空间不够的问题

单链表：优：动态开辟存储空间，方便插入和删除，长度不限，不浪费存储空间

缺：存取线性表中元素较慢

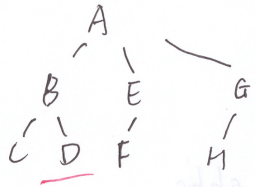
(1) 长度稳定，少插入和删除，并要快速存取线性表中元素

采用 顺序表 +6

(2) n个线性表并存，且处理中表的长度动态变化

采用 单链表

3.



+1

4.

三元组收序表

	行	列	数
#	1	3	2
	2	1	3
	3	3	1
	3	4	5
总	4	5	11

十字链表

行	1	2	3	3	
列	3	1	3	4	
数	2	3	1	5	

+2

5.

+2

T	a	a	b	a	a	b	c
next[j]	0	1	1	0	2	3	5
nextval	0	0	1	0	2	1	5

aabaabjc

	前	后
多	aabaa	abaab
	abaa	abaa
	baa	baab
	aab	aaab
		baa

abbja

前	后
ab, a	b, bb

aabajaa

前	后
a, aa, ab	a, ab, ba
aab	aba

aabaajb

	前	后
多	aaba	abaa
	aba	aba, baa

6.

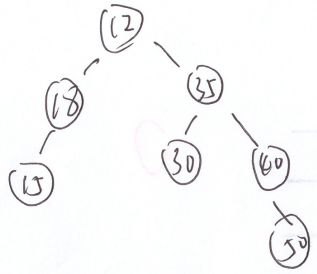
最早开始时间	最迟开始时间
$V_2 = 6$	$V_2 = 33$
$V_3 = 4$	$V_3 = 35$
$V_4 = 12$	$V_4 = 26$
$V_5 = 17$	$V_5 = a_1 + a_4 + a_5 + a_8 = 34$
$V_6 = 15$	$V_6 = a_1 + a_4 + a_3 + a_5 = 32$

AOE逐关键路径

1. ~~$V_1 \xrightarrow{a_1} V_2 \xrightarrow{a_4} V_6 \xrightarrow{a_6} V_5 \xrightarrow{a_8}$~~
1. $V_1 \xrightarrow{a_1} V_2 \xrightarrow{a_4} V_6 \xrightarrow{a_6} V_5 \xrightarrow{a_8} V_6 \xrightarrow{a_5} V_3 \xrightarrow{a_2} V_1$
2. $V_1 \to V_3 \to V_6 \to V_5 \to V_4 \to V_2 \to V_1$
3. $V_1 \to V_2 \to V_6 \to V_3 \to V_6 \to V_5$

~~4. V_1~~

7.



8. 简单

初始 4 1 3 5 2 7 6

1,4,3,5,2,7,6

1,3 4 5 2 7 6

1 3 4 2 5 7 6

1 3 2 4 5 7 6

1 2 3 4 5 7 6

1 2 3 4 5 6 7

快速

初始 4 1 3 5 2 7 6 +2

1 3 4 5 2 7 6

1 2 3 4 5 7 6

1 2 3 4 5 6 7

得 分	0

四、算法设计（每题 10 分，共 20 分）

1.

0

2.

0