# ① 当前作业

» 22级第六次 作业(查找与 排序)\_

» <u>22级第五次</u> 作业 (树)

**>> 2022级 (信** 息大类)数据 结构综合作业 (正确性和性 能)

» 2022级 (信 息大类)数据 结构综合作业 <u>(可扩展性)</u>

# り 历史作业

**>>** 22级第四次 作业 (栈和 队)\_\_\_

**>>** 22级第三次 作业 (线性 表)\_

>> 22级第二次 作业

**>>** 22级第一次 作业

» 21级第七次 作业 (图)

**>> 21级第六次** 作业(查找与 <u>排序)</u>

» 21级第五次 作业 (树)

**>> 2021级 (信** 息大类)数据 结构综合作业 (正确性和性

>> 2021级 (信 息大类)数据 结构综合作业 <u>(可扩展性)</u>

**>> 21级第四次** 作业 (栈和 队)

**>> 21级第三次** <u>作业</u>

21级第六次作业(查找与排序)

作业时间: 2022-05-19 18:00:00 至 2022-07-01 02:00:00

主要考查对查找和排序知识的掌握情况,请用相关知识完成本次作业。其中排序部分可以待最后一讲(排序部分)讲 完后再做。

● 选择题

2.

3.

4.

1.

首次提交时间:2022-05-19 19:48:08 最后一次提交时间:2022-05-19 19:48:09 对N 个元素的表做顺序查找时,若查找每个元素的概率相同,则平均查找长度为 A. N/2B. (N+1)/2C. N D. [(1+N)\*N]/2首次提交时间:2022-05-19 19:49:14 最后一次提交时间:2022-05-19 19:49:14 已知一个长度为16的顺序表L,其元素按关键字有序排列。若采用折半查找法查找一个L中不存在的元 素,则关键字比较次数最多为C A. 7 B. 6 C. 5 D. 4 首次提交时间:2022-05-19 19:49:42 最后一次提交时间:2022-05-19 19:49:45 对于长度为9的有序顺序表,若采用折半搜索,在等概率情况下搜索成功的平均搜索长度为 的值除以9。 A, 20 B、18 C、25 D、22 首次提交时间:2022-05-19 19:51:37 最后一次提交时间:2022-05-19 19:51:37 将数据元素2,4,6,8,10,12,14,16,18,20依次存放于一个一维数组中,然后采用折半查找方法查找元素12,被比较 过的数组元素的下标依次为〇 A. 10,16,12 B. 10,12,16 C. 4,7,5 D. 4,5,7

下面关于m 阶B- 树说法正确的是(B

①每个结点至少有两棵非空子树; ②树中每个结点至多有m-1个关键字;

首次提交时间:2022-05-19 19:52:30 最后一次提交时间:2022-05-19 19:52:30

③所有叶子在同一层上; ④当插入一个数据项引起B 树结点分裂后, 树长高一层。

A. 123 B.23 C.234 D.3

<u>≫ 21级第二次</u> 作业	6.	首次提交时间:2022-05-19 19:54:11 最后一次提交时间:2022-05-19 19:54:11
<u>&gt;&gt; 21级第一次</u> 作业		设有一组记录的关键字为{19, 14, 23, 1, 68, 20, 84, 27, 55, 11, 10, 79}, 用链地址法构造散列表,散列函数为H(key)=key MOD 13,散列地址为1的链中有 D
		A. 1 B. 2 C. 3 D. 4
	7.	首次提交时间:2022-05-19 19:55:22 最后一次提交时间:2022-05-19 19:55:22
		已知序列25,13,10,12,9是大顶堆,在序列尾部插入新元素18,将其再调整为大顶堆。调整过程中元素之间进行的比较次数是B
		A.1 B.2 C.4 D.5
	8.	首次提交时间:2022-05-19 19:55:43 最后一次提交时间:2022-05-19 19:55:43
		下列排序算法中(C)排序在一趟结束后不一定能选出一个元素放在其最终位
		置上。 A. 选择 B. 冒泡 C. 归并 D. 堆
	9.	首次提交时间:2022-05-19 19:56:02 最后一次提交时间:2022-05-19 19:56:02
		对一组数据(84, 47, 25, 15, 21)排序,数据的排列次序在排序的过程中的变化为 (1) 84 47 25 15 21 (2) 15 47 25 84 21 (3) 15 21 25 84 47 (4) 15 21 25 47 84 则采用的排序是 A
		A. 选择 B. 冒泡 C. 快速 D. 插入
	10.	首次提交时间:2022-05-19 19:56:54 最后一次提交时间:2022-05-19 19:56:54
		若要进行从小到大排序,数据元素序列 <b>11,12,13,7,8,9,23,4,5</b> 是采用下列排序方法之一得到的第二趟排序后的结果,则该排序算法只能是B
		A: 冒泡排序 B:插入排序 C:选择排序 D。二路归并排序
	11.	首次提交时间:2022-05-19 19:58:43 最后一次提交时间:2022-05-19 19:58:45
		选择:对有8个元素的序列(49,38,65,97,76,13,27,50)按从小到大顺序进行排序, A
		A.13,38,65,97,76,49,27,50
		B.13,27,38,49,50,65,76,97
		C.97,76,65,50,49,38,27,13
		D.13,38,65,50,76,49,27,97
	12.	首次提交时间:2022-05-19 19:59:08 最后一次提交时间:2022-05-19 19:59:08
		为实现快速排序算法,待排序序列宜采用的存储方式是 <u>A</u>
		A. 顺序存储 B. 散列存储 C. 链式存储 D. 索引存储

	快速排序在平均情况下的时间复杂度为_A	在最环情况下的时间复杂度为	
	A O(nlogn)		
	B. O(n <sup>2</sup> )		
	C.O(n <sup>2</sup> logn)		
	D. O(n)		
14.	首次提交时间:2022-05-19 19:59:30 最后一次提交时间:2022-05-19 19	9:59:30	
	给出一组关键字序列{12,2,16,30,8,28,4,10,20,6,18}, 当用快速排序 采用教材P336-337描述的算法)从小到大进行排序第一趟结束时的		
	A. 6, 2, 8, 10, 4, 12, 28, 30, 16, 20, 18		
	B. 6, 4, 8, 10, 2, 12, 28, 30, 16, 20, 18		
	C. 4, 2, 6, 10, 8, 12, 28, 30, 20, 16, 18 D. 4, 2, 8, 10, 6, 12, 16, 20, 28, 30, 18		
15.	首次提交时间:2022-05-19 20:00:07 最后一次提交时间:2022-05-19 20	0:00:07	
	若利用快速排序算法进行从小到大排序,下列选项中,不可能是经置后的排序结果的是 <sub>C</sub>	过两次选择分界元素并确定其最终位	
	A. 2,3,5,4,6,7,9		
	B. 2,7,5,6,4,3,9		
	C. 3,2,5,4,7,6,9		
	D. 4,2,3,5,7,6,9		
	_ · · ·,_,-,-,-,-		
/ 填3	<b>三</b> 题		
1.	首次提交时间:2022-05-19 20:00:36 最后一次提交时间:2022-05-19 20	0:00:36	N
	对序列(49,38,65,97,76,13,47,50)采用折半插入排序法进行排序,若把第列中,为寻找插入的合适位置需要进行 <u>3</u> 次元素间的比较。	7个元素47插入到已排序序	
2.	首次提交时间:2022-05-19 20:00:56 最后一次提交时间:2022-05-19 20	0:01:21	Z
	在有序表(k1,k2,,k99)中采用折半查找方法查找99次,其中至少有一个是是 <u>k50</u> 。(答案格式如:k99)	元素被比较了99次,该元素	
3.	首次提交时间:2022-05-19 20:01:23 最后一次提交时间:2022-05-19 20	0:01:23	N
	折半查找过程可以利用一棵称之为"判定树"的二叉树来描述。在长度为应判定树的根结点右孩子的值(某元素在序列中的位置)是 <u>9</u> 。	12的序列中进行折半查找对	
4.	首次提交时间:2022-05-19 20:02:43 最后一次提交时间:2022-05-26 19	9:21:03	Ž.
	若一个待散列存储的线性表为K=(18,25,63,50,42,32,9,45),散列函数为H8发生冲突的元素有 <u>3</u> 个。	H(k) = k MOD 9,则与元素1	

首次提交时间:2022-05-19 19:59:13 最后一次提交时间:2022-05-19 19:59:16

13.

5. 首次提交时间:2022-05-19 20:03:15 最后一次提交时间:2022-05-19 20:03:15

已提交

插入排序法的时间花费主要取决于元素间的比较次数,若具有n个元素的序列初始时已经是一个**递增**序列,则排序过程中一共要进行<u>n-1</u>次比较。

● 编程题

# 题目 分值 批阅信息

#### 1. 单词查找(查找-基本题)

### 【问题描述】

从标准输入中读入一个英文单词及查 找方式,在一个给定的英文常用单词 字典文件dictionary3000.txt中查找 该单词,返回查找结果(查找到返回 1,否则返回0)和查找过程中单词的 比较次数。查找前,先将所有字典中 单词读入至一个单词表(数组)中, 然后按要求进行查找。字典中单词总 数不超过3500,单词中的字符都是英 文小写字母,并已按字典序排好序 (可从课件下载区下载该字典文 件)。字典中的单词和待查找单词的 字符个数不超过20。

查找方式说明:查找方式以1<sup>4</sup>数字表示,每个数字含义如下:

- 1: 在单词表中以顺序查找方式查找, 因为单词表已排好序,遇到相同的或 第一个比待查找的单词大的单词,就 要终止查找;
- 2: 在单词表中以折半查找方式查找; 3: 在单词表中通过索引表来获取单词 查找范围,并在该查找范围中以折半 方式查找。索引表构建方式为: 以26 个英文字母为头字母的单词在字典中 的起始位置和单词个数来构建索引 表,如:

字母	起始位置	单词个数
a	0	248
b	248	167
	•••	•••

该索引表表明以字母a开头的单词在单词表中的开始下标位置为0,单词个数为248。

4:按下面给定的hash函数为字典中单词构造一个hash表,hash冲突时按字典序依次存放单词。hash查找遇到冲突时,采用链地址法处理,在冲突链表中找到或未找到(遇到第一个比待查找的单词大的单词或链表结束)便结束查找。

/\* compute hash value for string \*/

#define NHASH 3001

#define MULT 37

unsigned int hash(char \*str)

{

unsigned int h=0;

char \*p;

## 40.00 下载源文件

#### 得分40.00 最后一次提交时间:2022-05-19 20:45:50

#### 共有测试数据:9

平均占用内存:1.228K 平均CPU时间:0.00409S 平均墙钟时间:0.00408S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确
测试数据6	完全正确
测试数据7	完全正确
测试数据8	完全正确
测试数据9	完全正确

详细 🛭

```
for(p=str; *p!='\0'; p++)
    h = MULT*h + *p;
return h % NHASH;
}
```

提示: hash表可以构建成指针数组, h ash冲突的单词形成一有序链表。

### 【输入形式】

单词字典文件dictionary3000.txt存放在当前目录下,待查找单词和查找方式从标准输入读取。待查找单词只包含英文小写字母,与表示查找方式的整数之间以一个空格分隔。

# 【输出形式】

将查找结果和单词比较次数输出到标准输出上,两整数之间以一个空格分隔。

# 【样例输入与输出】

单词字典文件dictionary3000.txt与课件下载中提供的相同,下面两列中,左侧为待查找单词与查找方式,右侧为对应的输出结果:

wins 1
0 3314
wins 2
0 12
wins 3
0 7

0 7

wins 4

0 2

yes 1

1 3357

yes 2

1 10

yes 3

1 4

yes 4

1 1

# 【样例说明】

wins在单词字典中不存在,4种查找方式都输出结果0,顺序查找、折半查找、索引查找和hash查找的单词比较次数分别为:3314、12、7和2次(winshhash位置与字典中physics和suggest相同)。

yes在单词字典中存在,4种查找方式都输出结果1,顺序查找、折半查找、索引查找和hash查找的单词比较次数分别为:3357、10、4和1。

# 【评分标准】

该题要求输出查找结果和查找过程中的单词比较次数,提交程序名为find.c。

# 题目 分值 批阅信息

### 2. 排座位 (简) a

【问题描述】某班级要进行期末考试, 准备考试时打乱座次,现已按照学号顺 序人工为学生随机安排了座位号,但其 中可能会出现漏排和重复安排座位的情 况。编写程序读入人工安排的考试座位 安排表T1,对安排情况进行检查,并对 漏排和重复安排座位的情况进行修正, 修正后,若学生人数为N,则每位学生考 试座位安排应在1~N之间,其中没有缺 号和重号。假设T1中学号信息不会出现 重复,同一座位号最多有两位同学的座 位号相同,并且座位号不会连续漏排; 初始考试座位安排表存放在当前目录下 的in.txt中,其中包括每位学生的学号、 姓名和座位号,要求修正后的考试座位 安排表输出到当前目录下的out.txt文件 中。程序检查座位号的规则如下: 1、首先对考试座位安排表T1按座位号从 小到大的顺序排序(原始考试安排可能 有座位号相同情况,座位号相同时则按 原始学号顺序排序),得到按座位号排 序的安排表T2;

- 2、对表T2从头开始检查漏排座位号情况:假设当前表中安排的最大座位号为M,取M和N的较小值Q;从1号开始检查,若某个小于等于Q的座位序号没有安排学生,则将表T2的最后学生的座位设置为该座位号;若存在多个漏排座位,则从表T2最后依次向前设置;
- 3、然后再检查表T2中重排座位号情况: 假设当前表中安排的最大座位号为m,将 座位号重复的、学号较大的学生的座位 号依次设置为m+1、m+2、m+3.....;
- 4. 将调整好的表T2按学号由小到大序排序后按输出格式要求输出至指定输出文件中。

# 【输入形式】

从标准输入中读入学生人数 (不超过100 的正整数)。

初始考试座位安排表存储在当前目录下的in.txt文件中,已按照学号由小到大的顺序分行存储每位学生座位信息,依次为学生学号(不超过8位的正整数)、姓名(由不超过20位的英文字母组成)和座位号(不超过100的正整数),各数据间以一个空格分隔。最后一个学生座位信息后有回车换行。

# 【输出形式】

按照学号由小到大的顺序将修正后的考 试座位安排表输出到当前目录下的out.txt 文件中,每行依次为学号、姓名和座位 号,各数据之间以一个空格分隔。

【样例输入】

24

假设当前目录下的in.txt文件内容如下: 18373001 ShiTian 7 18373002 WangLi 15 18373003 LiGuoHong 23

#### 20.00 下载源文件

#### 得分20.00 最后一次提交时间:2022-05-19 19:46:07

#### 共有测试数据:5

平均占用内存:1.227K 平均CPU时间:0.00426S 平均墙钟时间:0.00423S

测试数据	评判结果	
测试数据1	完全正确	
测试数据2	完全正确	
测试数据3	完全正确	
测试数据4	完全正确	
测试数据5	完全正确	



18373005 QianSanQiang 26

18373006 ZhangQiang 8

18373007 SunXiXi 2

18373010 LiXing 12

18373011 TangYing 20

18373012 YangYang 4

18373013 ZhaoGang 27

18373014 ZhouLiang 18

18373015 WuShuo 9

18373016 ZhengSiSi 13

18373017 WangGong 27

18373018 TianTian 21

18373020 LeiLei 16

18373021 ZhangLou 10

18373022 WangLei 17

18373025 SunTian 24

18373026 JinXiang 18

18373028 PangHong 11

18373029 GaoLiang 2

18373030 GaoHang 6

18373031 YangYang 22

### 【样例输出】

当前目录下的out.txt文件内容应为:

18373001 ShiTian 7

18373002 WangLi 15

18373003 LiGuoHong 19

18373005 QianSanQiang 5

18373006 ZhangQiang 8

18373007 SunXiXi 2

18373010 LiXing 12

18373011 TangYing 20

18373012 YangYang 4 18373013 ZhaoGang 3

18373014 ZhouLiang 18

18373015 WuShuo 9

18373016 ZhengSiSi 13

18373017 WangGong 1

18373018 TianTian 21

18373020 LeiLei 16

18373021 ZhangLou 10

18373022 WangLei 17 18373025 SunTian 14

18373026 JinXiang 24

18373028 PangHong 11

18373029 GaoLiang 23

18373030 GaoHang 6

18373031 YangYang 22

## 【样例说明】

初始考试座位安排表中有24位学生的排 位信息,正确情况下这些学生安排的座 位号应为1~24。初始人工安排的座位信 息有1、3、5、14和19号座位漏排,有 2、18和27号座位重复安排学生。先对漏 排的座位进行修正:已安排的最大座位 号为27, 学号为18373017和18373013的 学生分别安排了27号座位,按照漏排座 位修正规则, 先将18373017学生安排在1 号座位,再将18373013学生安排在3号座 位;同理分别将18373005学生安排在5号 座位, 将18373025学生安排在14号座 位, 18373003号学生安排在19号座位。 当前安排的最大座位号为22,还有2号和 18号座位重复,将2号重复的学号较大的 18373029号学生安排在23号座位,将18 号重复的学号较大的18373026号学生安

分值 批阅信息

排在24号座位。这样修正后按照学号由 小到大的顺序输出学生座位信息到out.txt 中。

【评分标准】

题目

按照要求修正学生座位信息,提交程序 文件名为seat.c。 # 题目 分值 批阅信息

### 3. 整数排序(排序-基本题)

# 【问题描述】

从标准输入中输入一组<u>互不相同的</u>整数 (个数不超过100)及排序方式,按照<mark>从小到大排序</mark>,输出按某种算法排序的结 果及元素的比较次数。

说明:排序方式为一个1~5的整数,分别表示:

- 1:选择排序,比较次数是指选择未排序部分的最小元素时的比较次数。
- 2: 冒泡排序, 比较次数是指相邻元素的 比较次数, 若某趟排序中没有进行数据 交换, 就认为排序结束。
- 3: 堆排序, 比较次数是指根元素调整过程中根元素与子树根结点的比较次数, 即下面算法中红色语句的执行次数:

```
void adjust(int k[ ],int i,int n)
{
  int j,temp;
  temp=k[i];
  j=2*i+1;
  while(j<n){
    if(j < n-1 && k[j] < k[j+1])
      j++;
    if(temp > = k[j])
      break;
    k[(j-1)/2]=k[j];
    j=2*j+1;
  }
  k[(j-1)/2] = temp;
}
4: 二路归并排序, 比较次数是指两组有
序数据合并成一组时的比较次数,即下
面算法中红色语句的执行次数 (注意:
调用 merge时,要使用上课讲的递归算
法):
void merge(int x[],int tmp[],int left,int le
ftend,int rightend)
{
  int i=left, j=leftend+1, q=left;
  while(i<=leftend && j<=rightend)
  {
    if(x[i] < =x[j])
```

tmp[q++]=x[i++];

## 40.00 下载源文件

#### 得分40.00 最后一次提交时间:2022-05-19 19:45:44

#### 共有测试数据:5

平均占用内存:1.229K 平均CPU时间:0.00463S 平均墙钟时间:0.00460S

测试数据	评判结果
测试数据1	完全正确
测试数据2	完全正确
测试数据3	完全正确
测试数据4	完全正确
测试数据5	完全正确



else

tmp[q++]=x[j++];

```
}
 while(i<=leftend)
   tmp[q++]=x[i++];
 while(j<=rightend)
   tmp[q++]=x[j++];
 for(i=left; i<=rightend; i++)</pre>
   x[i]=tmp[i];
}
5: 快速排序, 比较次数是指分界元素与
其它元素的比较次数,即下面算法中红
色语句的执行次数:
void quickSort(int k[ ],int left,int right)
{
 int i, last;
 if(left<right){
   last=left;
    for(i=left+1;i<=right;i++)</pre>
     if(k[i] < k[left])
       swap(&k[++last],&k[i]);
    swap(&k[left],&k[last]);
    quickSort(k,left,last-1);
    quickSort(k,last+1,right);
 【输入形式】
首先在屏幕上输入2个整数,分别表示待
排序的整数个数及排序方式, 然后在下
一行依次输入待排序的整数。各整数之
间都以一个空格分隔。
【输出形式】
先在一行上输出排序结果,各整数间以
一个空格分隔。然后在下一行上输出排
序过程中的元素比较次数。
【样例1输入】
20 1
38 356 98 -102 126 46 65 -9 100 0 21 2
90 8 18 12 78 16 189 23
【样例1输出】
-102 -9 0 2 8 12 16 18 21 23 38 46 65 78
90 98 100 126 189 356
190
```

## 【样例1说明】

输入了20个整数数据,要求按照选择排序算法对输入的数据进行从小到大排序,输出排序结果,排序过程中元素的比较次数为190次。

# 【其它样例说明】

若输入的待排序数据与样例1完全相同, 要求的排序算法不同,则输出的排序结 果与样例1完全一样,但比较次数不同, 为了方便说明,下面左侧为排序方式, 右侧为对应的比较次数:

- 2 162
- 3 58
- 4 66
- 5 75

## 【评分标准】

该题要求按照指定算法对输入的数据进行排序,提交程序名为sort.c。

北京航空航天大学

若重置密码,请与当前的任课教师联系