

习题九 排序

一、单项选择题

1. 下列内部排序算法中:

- A. 快速排序
- B. 直接插入排序
- C. 二路归并排序
- D. 简单选择排序
- E. 起泡排序
- F. 堆排序

(1) 其比较次数与序列初态无关的算法是 ()

(2) 不稳定的排序算法是 ()

(3) 在初始序列已基本有序(除去 n 个元素中的某 k 个元素后即呈有序, $k \ll n$) 的情况下, 排序效率最高的算法是 ()

(4) 排序的平均时间复杂度为 $O(n \cdot \log n)$ 的算法是 () 为 $O(n \cdot n)$ 的算法是 ()

2. 比较次数与排序的初始状态无关的排序方法是 ()。

- A. 直接插入排序
- B. 起泡排序
- C. 快速排序
- D. 简单选择排序

3. 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. 选择
- B. 冒泡
- C. 快速
- D. 插入

4. 下列排序算法中 () 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

5. 一组记录的关键码为 (46, 79, 56, 38, 40, 84), 则利用快速排序的方法, 以第一个记录为基准得到的一次划分结果为 ()。

A. (38, 40, 46, 56, 79, 84) B. (40, 38, 46, 79, 56, 84)

C. (40, 38, 46, 56, 79, 84) D. (40, 38, 46, 84, 56, 79)

6. 下列排序算法中, 在待排序数据已有序时, 花费时间反而最多的是 () 排序。

- A. 冒泡
- B. 希尔
- C. 快速
- D. 堆

7. 就平均性能而言, 目前最好的内排序方法是 () 排序法。

- A. 冒泡
- B. 希尔插入
- C. 交换
- D. 快速

8. 下列排序算法中, 占用辅助空间最多的是: ()

- A. 归并排序
- B. 快速排序
- C. 希尔排序
- D. 堆排序

9. 若用冒泡排序方法对序列 {10, 14, 26, 29, 41, 52} 从大到小排序, 需进行 () 次比较。

- A. 3
- B. 10
- C. 15
- D. 25

10. 快速排序方法在 () 情况下最不利于发挥其长处。

- A. 要排序的数据量太大
- B. 要排序的数据中含有多个相同值
- C. 要排序的数据个数为奇数
- D. 要排序的数据已基本有序

11. 下列四个序列中, 哪一个是堆 ()。

A. 75, 65, 30, 15, 25, 45, 20, 10 B. 75, 65, 45, 10, 30, 25, 20, 15

C. 75, 45, 65, 30, 15, 25, 20, 10 D. 75, 45, 65, 10, 25, 30, 20, 15

12. 有一组数据 (15, 9, 7, 8, 20, -1, 7, 4), 用堆排序的筛选方法建立的初始堆为 ()

A. -1, 4, 8, 9, 20, 7, 15, 7

B. -1, 7, 15, 7, 4, 8, 20, 9

C. -1, 4, 7, 8, 20, 15, 7, 9

D. A, B, C 均不对。

二、填空题

1. 若待排序的序列中存在多个记录具有相同的键值, 经过排序, 这些记录的相对次序仍然保持不变, 则称这种排序方法是_____的, 否则称为_____的。

2. 按照排序过程涉及的存储设备的不同, 排序可分为_____排序和_____排序。

3. 直接插入排序用监视哨的作用是_____。

4. 对 n 个记录的表 $r[1..n]$ 进行简单选择排序, 所需进行的关键字间的比较次数为_____。

5. 下面的 c 函数实现对链表 head 进行选择排序的算法, 排序完毕, 链表中的结点按结点值从小到大链接。请在空框处填上适当内容, 每个空框只填一个语句或一个表达式:

```
#include <stdio.h>
typedef struct node {char data; struct node *link; }node;
node *select(node *head)
{node *p,*q,*r,*s;
 p=(node *)malloc(sizeof(node));
 p->link=head; head=p;
 while(p->link!=null)
 {q=p->link; r=p;
  while ((1)_____)
  { if (q->link->data<r->link->data) r=q;
    q=q->link;
  }
  if ((2)_____) {s=r->link; r->link=s->link;
                s->link= ((3)____); ((4)____);}
  ((5)____) ;
 }
 p=head; head=head->link; free(p); return(head);
}
```

6. 下面的排序算法的思想是: 第一趟比较将最小的元素放在 $r[1]$ 中, 最大的元素放在 $r[n]$ 中, 第二趟比较将次小的放在 $r[2]$ 中, 将次大的放在 $r[n-1]$ 中, ..., 依次下去, 直到待排序列为递增序。(注: \leftrightarrow 代表两个变量的数据交换)。

```
void sort(Sqlist &r,int n) {
    i=1;
    while((1)____) {
        min=max=1;
        for (j=i+1; (2)____ ;++j)
            {if((3)____) min=j; else if(r[j].key>r[max].key) max=j; }
        if((4)____) r[min] < ---- >r[j];
        if(max!=n-i+1){if ((5)____) r[min] < ---- > r[n-i+1]; else ((6)____); }
        i++;
    }
}
```

```
}//sort
```

7. 下列算法为奇偶交换排序, 思路如下: 第一趟对所有奇数的 i , 将 $a[i]$ 和 $a[i+1]$ 进行比较, 第二趟对所有偶数的 i , 将 $a[i]$ 和 $a[i+1]$ 进行比较, 每次比较时若 $a[i] > a[i+1]$, 将二者交换; 以后重复上述二趟过程, 直至整个数组有序。

```
void oesort (int a[n])
{int flag, i, t;
  do {flag=0;
    for(i=1; i<n; i++, i++)
      if(a[i]>a[i+1])
        {flag=(1)_____; t=a[i+1]; a[i+1]=a[i]; (2)_____;}
    for (3)_____
      if (a[i]>a[i+1])
        {flag=(4)_____; t=a[i+1]; a[i+1]=a[i]; a[i]=t;}
  }while (5)_____;
}
```

第九章 排序

一、单项选择题

1. (1) DC (2) ADF (3) B (4) ACF BDE
2. D
3. A
4. C
5. C
6. C
7. D
8. A
9. C
10. D
11. C
12. C

二、填空题

1. 稳定、不稳定
2. 内部、外部
3. 免去查找过程中每一步都要检测整个表是否查找完毕, 提高了查找效率。
4. $n(n-1)/2$
5. 题中为操作方便, 先增加头结点 (最后删除), p 指向无序区的前一记录, r 指向最小值结点的前驱, 一趟排序结束, 无序区第一个记录与 r 所指结点的后继交换指针。
(1) $q \rightarrow \text{link} \neq \text{NULL}$ (2) $r \neq p$ (3) $p \rightarrow \text{link}$ (4) $p \rightarrow \text{link} = s$ (5) $p = p \rightarrow \text{link}$
6. (1) $i < n-i+1$ (2) $j < n-i+1$ (3) $r[j].\text{key} < r[\text{min}].\text{key}$ (4) $\text{min} \neq i$ (5) $\text{max} == i$
(6) $r[\text{max}] \leftrightarrow r[n-i+1]$
7. (1) 1 (2) $a[i] = t$ (3) $(i=2; i \leq n; i+=2)$ (4) 1 (5) flag