

- (6) 抽象数据类型。
- (7) 数据结构的形式化定义、数据结构两个构成要素的含义。
- (8) 算法的特性及算法设计要求。
- (9) 算法的时间复杂度。
- (10) 函数参数传递的主要方式及特点。

2. 选择题:

(1) 设某数据结构的二元组形式表示为  $A = (D, R)$ ,  $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ,  $R = \{r \mid r = \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 1, 4 \rangle, \langle 2, 5 \rangle, \langle 2, 6 \rangle, \langle 3, 7 \rangle, \langle 3, 8 \rangle, \langle 3, 9 \rangle\}$ , 则数据结构  $A$  是 ( B )。

- A. 线性结构      B. 树形结构      C. 物理结构      D. 图结构

(2) 数据的最小单位是 ( A )。

- A. 数据项      B. 数据类型      C. 数据元素      D. 数据变量

(3) 若需要利用形式参数直接访问修改实参值, 则应将形参说明为 ( A ) 参数。

- A. 指针      B. 值参数      C. 数值      D. 整型

(4) 下列程序段的时间复杂度为 ( A )。

$i = 0, s = 0; \text{ while } (s < n) \{ s = s + 1; i++; \}$

- A.  $O(n^{1/2})$       B.  $O(n^{1/3})$       C.  $O(n)$       D.  $O(n^2)$

$$S = 1 + 2 + \dots + m = \frac{m(m+1)}{2} < n$$

m 为常数, 次数

(5) 执行下面程序段时, 语句 S 的执行次数为 ( C )。

$\text{for } (i = 0; i < n; i++)$   
 $\text{for } (j = 0; j < i; j++)$   
 S;

$$\sum_{i=0}^n (i+1) = \frac{(n+1)(n+2)}{2}$$

- A.  $n^2$       B.  $n^2/2$       C.  $n(n+1)$       D.  $n(n+1)/2$

3. 计算下列程序段中  $x = x + 1$  的语句频度:

$\text{for } (i = 1; i \leq n; i++)$   
 $\text{for } (j = 1; j \leq i; j++)$   
 $\text{for } (k = 1; k \leq j; k++)$   
 $x = x + 1;$

$$\sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^j 1 = \sum_{i=1}^n \sum_{j=1}^i j = \sum_{i=1}^n \frac{i(i+1)}{2} = \frac{n(n+1)(n+2)}{6}$$

$$= \sum_{i=1}^n \frac{i(i+1)}{2}$$

4. 编写算法, 求一元多项式  $P_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n$  的值  $P_n(x_0)$ , 并确定算法中每一语句的执行次数和整个算法的时间复杂度, 要求时间复杂度尽可能小, 算法中不能使用求幂函数。注意: 本题中的输入为  $a_i (i = 0, 1, \dots, n)$ ,  $x$  和  $n$ , 输出为  $P_n(x_0)$ 。通常算法的输入和输出可采用下列两种方式之一:

(1) 通过参数表中的参数显式传递。

(2) 通过全局变量隐式传递。

试讨论这两种方法的优缺点, 并在本题算法中以你认为较好的一种方式实现输入和输出。

5. 函数调用结果带出方式。

## 实 习 题 1

设计实现抽象数据类型“有理数”, 基本操作包括有理数的加法、减法、乘法、除法, 以及求有理数的分子、分母。

(2) 单链表的优缺点。

(3) 单链表中的头指针、头结点和首元素结点。

2. 选择题:

(1) 某线性表中最常用的操作是存取序号为  $i$  的元素和在最后进行插入和删除运算, 则采用 (D) 存储方式时间性能最好。

- A. 双向链表      B. 双向循环链表      C. 单向循环链表      D. 顺序表

(2) 下列选项中, (D) 是链表不具有的特点。

- A. 插入和删除运算不需要移动元素      B. 所需要的存储空间与线性表的长度成正比  
C. 不必事先估计存储空间大小      D. 可以随机访问表中的任意元素

(3) 在链表中最常用的操作是删除表中最后一个结点和在最后一个结点之后插入元素, 则采用 (D) 最节省时间。

- A. 头指针的单向循环链表      B. 双向链表  
C. 带尾指针的单向循环链表      D. 带头指针双向循环链表

(4) 下面关于线性表的叙述错误的是 (D)。

- A. 线性表采用顺序存储必须占用一片连续的存储空间  
B. 线性表采用链式存储不必占用一片连续的存储空间  
C. 线性表采用链式存储便于插入和删除操作的实现  
D. 线性表采用顺序存储便于插入和删除操作的实现

(5) 设一条单链表的头指针变量为 head 且该链表没有头结点, 则其判空条件是 (A)。

- A.  $head == 0$       B.  $head \rightarrow next == 0$       C.  $head \rightarrow next == head$       D.  $head != 0$

(6) 设带有头结点的单向循环链表的头指针变量为 head, 则其判空条件是 (C)。

- A.  $head == NULL$       B.  $head \rightarrow next == NULL$       C.  $head \rightarrow next == head$       D.  $head != NULL$

(7) 设顺序线性表中有  $n$  个数据元素, 则删除表中第  $i$  个元素需要移动 (A) 个元素。

- A.  $n-i$       B.  $n+1-i$       C.  $n-1-i$       D.  $i$

(8) 有一个含头结点的双向循环链表, 头指针为 head, 则其为空的条件是 (C)。

- A.  $head \rightarrow prior == NULL$       B.  $head \rightarrow next == NULL$   
C.  $head \rightarrow next == head$       D.  $head \rightarrow next \rightarrow prior == NULL$

(9) 在一个单链表中, 删除指针  $p$  所指的后继结点, 以下语句正确的是 (D)。

- A.  $p \rightarrow next = p \rightarrow next \rightarrow next$ ;  $free(p \rightarrow next)$ ;      B.  $free(p \rightarrow next)$ ;  $p \rightarrow next = p \rightarrow next \rightarrow next$ ;  
C.  $p = p \rightarrow next$ ;      D.  $s = p \rightarrow next$ ;  $p \rightarrow next = s \rightarrow next$ ;  $free(s)$ ;

(10) 设某顺序表中第一个元素的地址是 Base, 每个结点占  $m$  个单元, 则第  $i$  个结点的地址为 (A)。

- A.  $Base + (i-1) \times m$       B.  $Base + i \times m$       C.  $Base - i \times m$       D.  $Base + (i+1) \times m$

3. 已知顺序表  $L$  递增有序, 写一算法, 将元素  $X$  插入线性表的适当位置, 以保持线性表的有序性。

4. 编写算法, 从顺序表中删除自第  $i$  个元素开始的  $k$  个元素。

5. 已知两个递增有序的顺序表  $La$  和  $Lb$ , 编写算法将  $La$  和  $Lb$  合并成一个递减有序的顺序表, 并将合并结果存放在  $La$  中 (假设表  $La$  空间足够大, 不会溢出)。要求: 时间复杂度为  $O(n)$ 。

6. 某顺序表  $L$  中存放整型数据, 编写算法, 在时间复杂度  $O(n)$ 、空间复杂度  $O(1)$  内, 对表  $L$  进行重新排序, 将奇数全部放在前面, 偶数全部放在后面。

### 习 题 3

1. 简述下列术语或概念:

- (1) 递归进层需要做的三件事。
  - (2) 递归退层需要做的三件事。
  - (3) 顺序队列的“假溢出”现象。
  - (4) 循环队列的判空、判满条件(少用一个空间区分队空队满)。
2. 选择题

- (1) 输入序列为 123, 若进栈、出栈操作可以交替进行, 则不能得到的出栈序列是( **B** )。
- A. 321                      B. 312                      C. 123                      D. 132

- (2) 以下会用到栈的应用是( **D** )。

- A. 递归                      B. 子程序调用  
C. 括号匹配                      D. 以上选项均是

- (3) 栈和队列的共同点是( **C** )。

- A. 都是先进先出                      B. 都是先进后出  
C. 只允许在端点处插入和删除元素                      D. 它们没有共同点

- (4) 循环队列存储在数组  $A[0..m-1]$  中, 则入队时 rear 应变化为( **C** )。

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

- (5) 设有一个顺序共享栈  $S[0..n-1]$ , 其中第一个栈顶指针 top1 的初值为 -1, 第二个栈顶指针 top2 的初值为 n, 则判断共享栈满的条件是( **C** )。

- A. top1 == top2                      B. top1 + top2 == n                      C. top1 + 1 == top2                      D. top1 - 1 == top2

3. 按照四则运算加、减、乘、除和幂运算(^)优先关系的惯例, 画出对下列算术表达式求值时运算数栈和运算符栈的变化过程:

$$A - B * C / D + E * F$$

4. 已知表达式为  $a \times b + (c - d / e) \times f$ :

- (1) 编写算法, 将原表达式转换为后缀表达式  $ab \times cde / - f \times +$ 。

- (2) 编写算法, 对转换后的后缀表达式进行求值。

5. 假设以带头结点的循环链表表示队列, 并且只设一个指针指向队尾元素结点(注意不设头指针), 试编写相应的队列初始化、入队列和出队列算法。

6. 要求循环队列不损失一个空间全部都能得到利用, 设置一个标志域 tag, 以 tag 为 0 或 1 来区分头尾指针相同时的队列状态的空与满, 试编写与此结构相应的入队与出队算法。

7. 设有 4 个元素 1、2、3、4 依次进栈, 而出栈操作可随时进行(进出栈可任意交错进行, 但要保证进栈次序不破坏 1、2、3、4 的相对次序), 试写出所有不可能的出栈次序和所有可能的出栈次序。

8. 已知递归函数如下:

综上所述,上三角矩阵  $A_{n \times n}$  中任意元素  $a_{ij}$  与  $B[k]$  的对应关系为

$$\begin{cases} k=f(i,j)=m, & i>j \\ k=f(i,j)=\frac{(i-1)(2n-i+2)}{2}+j-i+1, & i \leq j \end{cases}$$

例 5.5 求广义表表头与表尾。

已知广义表  $L=((x,y,z),a,(u,t,w))$ , 求  $\text{Head}(\text{Head}(\text{Tail}(\text{Tail}(L))))$  的结果。

解:

- ① 用 Tail 运算去掉表头  $(x,y,z)$ , 即  $\text{Tail}(L)=(a,(u,t,w))$ ;
- ② 再用 Tail 运算去掉表头  $a$ , 即  $\text{Tail}(\text{Tail}(L))=((u,t,w))$ ;
- ③ 用 Head 运算取出表头  $(u,t,w)$ , 即  $\text{Head}(\text{Tail}(\text{Tail}(L)))=(u,t,w)$ ;
- ④ 再用 Head 运算取出表头  $u$ , 即  $\text{Head}(\text{Head}(\text{Tail}(\text{Tail}(L))))=u$ 。

所以,  $\text{Head}(\text{Head}(\text{Tail}(\text{Tail}(L))))=u$ 。

## 习 题 5

### 1. 选择题

(1) 设有一个二维数组  $A[m][n]$ , 假设  $A[0][0]$  存放地址为 644,  $A[2][2]$  存放地址为 676, 每个元素占一个空间, 则  $A[3][3]$  的存放地址为 (C)。

A. 688

B. 678

C. 692

D. 696

(2) 设有一个 10 阶的下三角矩阵  $A$  (包括对角线), 按照从上到下、从左到右的顺序存储到 55 个连续的存储单元中, 每个数组元素占 1 个字节的存储空间, 则  $A[5][4]$  的地址与  $A[0][0]$  的地址之差为 (B)。

A. 10

B. 19

C. 28

D. 55

2. 设有一个上三角矩阵  $A_{n \times n}$ , 将其上三角中的元素逐列压缩存储到一个大小为  $n(n+1)/2$  的一维数组  $C$  中 (下标从 1 开始), 请给出计算上三角矩阵中任意元素  $a_{ij}$  ( $i < j$ ) 在一维数组  $C$  中位置的公式。

3. 设有三对角矩阵  $A_{n \times n}$ , 将其三条对角线上的元素逐行压缩存储到一个大小为  $3n-2$  的一维数组  $B$  中 (下标从 1 开始), 使得  $B[k]=a_{ij}$ , 求:

(1) 用  $i, j$  表示  $k$  的下标变换公式;

(2) 用  $k$  表示  $i, j$  的下标变换公式。

4.  $n$  阶对称矩阵  $A$  以行序为主序压缩存储在一维数组  $B$  中, 存储其下三角元素 (包括对角线), 计算  $A[i][j]$  与  $B[k]$  之间的对应关系。

5. 在算法 5.3 (稀疏矩阵一次定位快速转置算法) 中, 将计算  $\text{position}[\text{col}]$  的方法稍加改动, 使算法只占用一个辅助向量空间。

6. 编写算法, 实现三元组表表示的两个稀疏矩阵的加法。

7. 编写算法, 实现一个在十字链表中删除非零元素  $a_{ij}$ 。

8. 画出下列广义表的两种存储结构图示:

$((((a), b)), ((()d), (e, f)))$



## 习 题 6

## 1. 选择题

- (1) 对任意一棵树, 设它有  $n$  个结点, 这  $n$  个结点的度数之和为 ( )。
- A.  $n$                       B.  $n+1$                       C.  $n-1$                       D.  $2 \times n$
- (2) 某二叉树上有 50 个叶子结点, 则该二叉树的总结点数至少应有 ( ) 个。
- A. 100                      B. 99                      C. 98                      D. 101
- (3) 具有 100 个结点的完全二叉树, 若按从上到下、从左到右对其编号 (根结点为 1), 则编号最小的叶子结点的编号为 ( )。
- A. 100                      B. 50                      C. 51                      D. 49
- (4) 一棵二叉树的后序遍历序列为 DABEC, 中序遍历序列为 DEBAC, 则其先序遍历序列为 ( )。
- A. DEABC                      B. DECAB                      C. CEDBA                      D. ACBED
- (5) 树的路径长度是从树的根结点到每个结点的路径长度的 ( )。
- A. 总和                      B. 最大值                      C. 最小值                      D. 平均值
- (6) 以下几个符号的编码集合中, 是前缀码的是 ( )。
- A.  $\{0, 1, 01, 110, 1111\}$                       B.  $\{11, 10, 001, 101, 0001\}$
- C.  $\{00, 010, 0110, 1000, 1100\}$                       D.  $\{b, c, ab, ac, aba, abb, abc\}$
- (7) 下列存储形式中, ( ) 不是树的存储形式。
- A. 双亲表示法                      B. 孩子兄弟表示法
- C. 孩子链表表示法                      D. 顺序存储表示法

## 2. 填空题

- (1) 高度为  $h$  的二叉树, 最多有  $2^h - 1$  个结点, 最少有  $h$  个结点; 高度为  $h$  的完全二叉树, 最多有  $2^h - 1$  个结点, 最少有  $2^{h-1}$  个结点。
- (2)  $n$  个结点的  $k$  叉树, 若用具有  $k$  个 child 域的等长链结点存储树的每一个结点, 则空的 child 域有  $(nk + n + 1)$  个。
- (3) 一棵完全二叉树共有 520 个结点, 该完全二叉树共有  $260$  个叶子结点,  $(1)$  个度为 1 的结点,  $259$  个度为 2 的结点。
- (4) 已知完全二叉树的第 7 层有 10 个结点, 则整个二叉树的结点数为  $(73)$  个。
- (5) 某完全二叉树第 6 层有 10 个叶子结点, 则该完全二叉树最多有  $(107)$  个结点, 最少有  $(41)$  个结点。

## 3. 综合应用

- (1) 分别画出具有 3 个结点的树和 3 个结点的二叉树的所有不同形态。
- (2) 已知一棵度为  $k$  的树中有  $n_1$  个度为 1 的结点,  $n_2$  个度为 2 的结点,  $\dots, n_k$  个度为  $k$  的结点, 求该树中有多少个叶子结点。
- (3) 假设一棵二叉树的先序序列为 EBADCFHGKJ, 中序序列为 ABCDEFGHIJK, 试画出该二叉树。
- (4) 描述满足下列条件的二叉树:
- 先序遍历和中序遍历相同:  $DLR \quad LDR$

```

#define MAX 20 /* 定义最大顶点数目 */

int CenterVex (AdjMatrix G)
/* 在带权图 G 中求一个中心顶点, 返回该顶点编号 */
{
    int i, j, k;
    float min, len;
    float A[MAX][MAX]; /* A[i][j] 存放 i 和 j 之间的最短路径长度 */

    for (i=0; i<G.vexnum; i++) /* 初始化 A[i][j] */
        for (j=0; j<G.vexnum; j++)
            A[i][j] = G.arcs[i][j].adj;
    A[i][i] = 0; /* i 到 i 的路径长度为 0 */

    for (k=0; k<G.vexnum; k++) /* 求每一对顶点之间的最短路径长度 */
        for (i=0; i<G.vexnum; i++)
            for (j=0; j<G.vexnum; j++)
                if (A[i][k]+A[k][j]<A[i][j])
                    A[i][j] = A[i][k]+A[k][j];
    min=MAXLEN; k=0;
    for (i=0; i<G.vexnum; i++) /* 选择最短路径长度之和最小的顶点 vk */
        len=0;
        for (j=0; j<G.vexnum; j++) /* 求 vi 到其余顶点最短路径长度之和 */
            len=len+A[i][j];
        if (len<min)
            k=i; min=len;
    return(k);
}
    
```

### 习 题 7

#### 1. 选择题

- (1) 有向图  $G$  用邻接矩阵  $A$  存储, 则顶点  $i$  的入度等于  $A$  中 ( )。
- A. 第  $i$  行非无穷元素之和  
B. 第  $i$  列非无穷元素之和  
C. 第  $i$  行非零且非无穷元素个数  
D. 第  $i$  列非零且非无穷元素个数
- (2) 图的深度优先遍历算法类似于二叉树的 ( )。
- A. 先序遍历  
B. 中序遍历

#### D. 层次遍历

- (3) 关键路径是 AOE 网中从源点到汇点的 ( ) 路径。
- A. 最长  
B. 最短  
C. 所经边数最多  
D. 所经边数最少
- (4)  $n$  个顶点的无向连通图, 其至少有 ( ) 条边。
- A.  $n$   
B.  $n-1$   
C.  $n+1$   
D.  $2n$
- (5) 一个具有  $n$  个结点、 $k$  条边的非连通无向图是一个森林 ( $n>k$ ), 则该森林中有 ( ) 棵树。
- A.  $n$   
B.  $n-k$   
C.  $k$   
D.  $n+k$

#### 2. 填空题

- (1) 某有向图的所有顶点的出度之和是所有顶点的入度之和的 ( ) 倍。
- (2) 邻接表适合存储稀疏图, 邻接矩阵适合存储稠密图 (填写“稀疏”或“稠密”)。
- (3) 当顶点在图中的位置确定后, 一个图的邻接矩阵表示法是 ( ) 的, 而邻接表表示法是 ( ) 的 (填写“唯一”或“不唯一”)。
- (4) 用邻接矩阵存储有  $n$  个顶点和  $e$  条边的有向图, 确定某个顶点的出度的时间复杂度是 ( )。
- (5)  $n$  个顶点构成一个环, 该图共有 ( ) 棵生成树。

#### 3. 综合应用

- (1) 已知如图 7.28 所示的有向图, 请给出该图的:

- ① 每个顶点的入度、出度;
- ② 邻接矩阵;
- ③ 邻接表;
- ④ 强连通分量。

- (2) 已知某无向图的邻接矩阵存储结构如图 7.29 所示, 从顶点 1 开始, 分别给出其深度和广度优先遍历顶点序列和生成树。

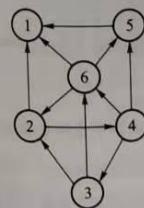


图 7.28 习题 3(1) 图

	1	2	3	4	5	6	7
1	∞	2	6	5	4	∞	∞
2	2	∞	∞	∞	∞	∞	4
3	6	∞	∞	∞	∞	2	∞
4	5	∞	∞	∞	∞	∞	3
5	4	∞	2	∞	∞	3	∞
6	∞	∞	∞	∞	5	∞	2
7	∞	4	∞	∞	∞	2	∞

图 7.29 习题 3(2) 图

- (3) 图 7.30 是有向图  $G$  的邻接表表示法。从结点  $v_1$  出发, 分别给出其深度和广度优先遍历顶点序列和生成树。

## 习 题 8

## 1. 选择题

(1) 对长度为 4 的元素, 从前往后顺序查找。如果查找第一个元素的概率是  $1/2$ , 查找第二个元素的概率是  $1/3$ , 查找第三个元素的概率是  $1/8$ , 查找第四个元素的概率是  $1/24$ 。则查找任意元素的平均查找长度为 (B)。

- A.  $5/2$       B.  $41/24$       C.  $73/24$       D. 1

(2) 对 1 000 个有序顺序表进行折半查找, 查找到存在的元素最多需要 (C) 次关键字比较。

- A. 8      B. 9      C. 10      D. 11

(3) 设顺序存储的线性表共 123 个元素, 按分块查找的要求等分为 3 块, 若对索引表采用顺序查找来确定块, 并在确定的块中进行顺序查找, 则在查找概率相等的情况下, 查找成功时的平均查找长度为 (B)。

- A. 21      B. 23      C. 41      D. 62

(4) 折半查找过程所对应的判定树一定是一棵 (C)。

- A. 完全二叉树      B. 满二叉树  
C. 平衡二叉树      D. 只有度为 0 和 2 的二叉树

(5) 对于下列关键字序列, 不可能构成某二叉排序树的查找路径的是 (C)。

- A. 85, 12, 80, 22, 78, 60      B. 102, 30, 100, 44, 90, 55  
C. 15, 90, 70, 99, 46, 48      D. 22, 30, 77, 68, 45, 50

## 2. 填空题

(1) 折半查找的前提条件是 (有序) 和 (有序)。

(2) 从空树开始, 依次插入元素 | 33, 22, 14, 56, 40, 78, 10, 19 | 构成一棵二叉排序树。若查找每个元素的概率均相同, 则该二叉排序的平均查找长度为 (11/4)。

(3) 对有序表 | 10, 22, 41, 106, 113, 167, 188, 190, 205 | 进行折半查找, 查找 188 需要进行 (2) 次关键字比较, 查找 100 需要进行 (4) 次关键字比较。

(4) 对二叉排序树进行中序遍历, 能得到一个有序序列。

(5) 平衡因子的取值为 (0, -1, 1)。

## 3. 综合应用

(1) 画出对长度为 10 的有序表进行折半查找的判定树, 并求其等概率时查找成功和不成功的平均查找长度。

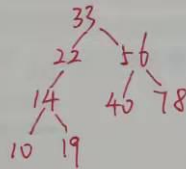
(2) 已知长度为 12 的表 (Jan, Feb, Mar, Apr, May, June, July, Aug, Sep, Oct, Nov, Dec)。

① 试按表中元素的顺序依次插入一棵初始为空的二叉排序树, 画出插入完成后的二叉排序树, 并求其等概率情况下查找成功和不成功的平均查找长度。

② 按表中元素的顺序依次构造一棵平衡二叉排序树, 并求其等概率情况下查找成功的平均查找长度。

(3) 选取哈希函数  $H(k) = (3k) \% 11$ , 用线性探测再散列处理冲突。试在 0~10 的散列地址空间中, 对关键字序列 (22, 41, 53, 46, 30, 13, 01, 67) 构造哈希表, 并求等概率情况下查找成功与不成功时的平均查找长度。

(4) 假设哈希函数为  $h(\text{key}) = (\text{key 值的首字母序号}) \% p$ , 其中  $p$  为小于等于哈希表长度的最大素数, 用线性探测再散列处理冲突。试为下列关键字建立一个装填因子不小于 0.75 的哈希表, 并计算等概率情况下查找



假设有 300 个记录,其关键字均为小于 1 000 的正整数,并且互不相等。试设计一种排序方法,以尽可能少的比较次数和移动次数实现排序。

解:由于题目中没有限制辅助空间,因此可以采用哈希表方法在  $O(n)$  时间复杂度内实现辅助存放。由于是按地址与内容对应地进行存储,可通过按序将结果另行置入结果数组的方式实现排序。

#### 【算法思想】

- ① 设计辅助数组  $b[1..999]$ ,每个元素存放一个记录,初始时全部置为空记录。
- ② 逐一扫描记录序列  $r[1..300]$  的每一个记录,若记录  $r[i]$  的关键字为  $K$ ,则将其放到  $b[K]$  中。
- ③ 再依次将  $b[1..999]$  中的非空记录逐一按顺序存于  $r[1..300]$  中。

### 习 题 9

#### 1 选择题

- (1) 对同一序列分别进行直接插入排序和折半插入排序,两者之间的区别是( C )。
    - A. 排序趟数不同
    - B. 元素移动次数不同
    - C. 比较次数不同
    - D. 使用辅助空间不同
  - (2) 以下哪种排序算法可能会出现:不完成最后一趟排序,所有元素可能都不在其最终位置?( C )
    - A. 堆排序
    - B. 冒泡排序
    - C. 插入排序
    - D. 快速排序
  - (3) 对一组数据  $\{34, 21, 55, 10, 67, 28\}$  进行排序,前三趟的排序结果如下。
 

第一趟:21, 34, 10, 55, 28, 67  
 第二趟:21, 10, 34, 28, 55, 67  
 第三趟:10, 21, 28, 34, 55, 67

 则该排序方法可能是( B )。
    - A. 堆排序
    - B. 冒泡排序
    - C. 插入排序
    - D. 快速排序
  - (4) 一组记录的关键字为  $\{46, 79, 56, 38, 40, 84\}$ ,利用快速排序法,以第一个记录为基准得到的一次划分结果为( C )。
    - 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
  - (5) 下列排序算法中,( C )算法可能会出现下述情况:初始数据有序时,花费时间反而最多。
    - A. 堆排序
    - B. 冒泡排序
    - C. 快速排序
    - D. 希尔排序
  - (6) 下列序列中,可看作是堆的是( C )。
    - A.  $\{80, 70, 35, 20, 30, 50, 25, 15\}$
    - B.  $\{80, 70, 50, 15, 35, 30, 25, 20\}$
    - C.  $\{80, 50, 70, 35, 20, 30, 25, 15\}$
    - D.  $\{80, 50, 70, 15, 30, 35, 25, 20\}$
2. 填空题
- (1) 对关键字序列  $\{44, 34, 66, 76, 12, 58, 91, 22\}$  采用快速排序法进行排序,以第一个元素为基准,一趟排序

的结果是( )。

- $\{22, 34, 12, 34, 44, 76, 58, 91, 66\}$
- (2) 快速排序在最坏情况下的时间复杂度为(  $O(n^2)$  )。
  - (3) 时间复杂度为  $O(n^2)$ ,空间复杂度为  $O(1)$ ,排序性能与初始序列的顺序无关的排序算法是( 简单选择 );时间复杂度为  $O(n \log n)$ ,空间复杂度为  $O(n)$ ,排序性能与初始序列的顺序无关的排序算法是( 归并 );
  - (4) 直接插入排序、冒泡排序、简单选择排序中,不稳定的排序算法是( 简单选择 );
  - (5) 希尔排序、快速排序、堆排序和归并排序中,稳定的排序算法是( 归并 )。

#### 3. 综合应用

(1) 以关键字序列  $\{503, 087, 512, 061, 908, 170, 897, 275, 653, 426\}$  为例,手工执行以下排序算法,写出每一趟排序结束时的关键字状态。

- ① 直接插入排序;
- ② 希尔排序(增量序列为  $5, 3, 1$ );
- ③ 快速排序;
- ④ 堆排序;
- ⑤ 归并排序;
- ⑥ 基数排序。

(2) 已知一组关键字  $\{40, 27, 28, 12, 15, 50, 7\}$ ,要求采用快速排序法从小到大排序。请写出每趟排序后的划分结果。

(3) 已知一组关键字  $\{40, 27, 28, 12, 15, 50, 7\}$ ,要求采用堆排序从小到大排序:

- ① 请画出建立初始堆的过程图示;
  - ② 请画出前三趟堆排序的过程图示。
- (4) 阅读下列排序算法,并与已学算法相比较,讨论算法中基本操作的执行次数。

void sort (SqList &r, int n)

```

{
    i = 1;
    while (i < n - i + 1) {
        min = max = i;
        for (j = i + 1; j <= n - i + 1; j++) {
            if (r[j].key < r[min].key) max = j;
            else if (r[j].key > r[max].key) max = j;
        }
        if (min != i) { w = r[min]; r[min] = r[i]; r[i] = w; }
        if (max != n - i + 1) {
            if (max == i) { w = r[min]; r[min] = r[n - i + 1]; r[n - i + 1] = w; }
            else { w = r[max]; r[max] = r[n - i + 1]; r[n - i + 1] = w; }
        }
        i++;
    }
}

```

- (5) 奇偶交换排序如下所述:第一趟对所有奇数  $i$ ,将  $a[i]$  和  $a[i+1]$  进行比较;第二趟对所有偶数  $i$ ,将  $a[i]$