

## 南信大数据结构期中模拟试卷（一）

### 一、选择题（20 分）

- 组成数据的基本单位是（ 1.C ）。  
(A) 数据项 (B) 数据类型 (C) 数据元素 (D) 数据变量
- 设数据结构  $A=(D, R)$ ，其中  $D=\{1, 2, 3, 4\}$ ， $R=\{r\}$ ， $r=\{<1, 2>, <2, 3>, <3, 4>, <4, 1>\}$ ，则数据结构 A 是（ C ）。  
(A) 线性结构 (B) 树型结构 (C) 图型结构 (D) 集合
- 数组的逻辑结构不同于下列（D）的逻辑结构。  
(A) 线性表 (B) 栈 (C) 队列 (D) 树
- 二叉树中第  $i (i \geq 1)$  层上的结点数最多有（C）个。  
(A)  $2i$  (B)  $2^i$  (C)  $2^{i-1}$  (D)  $2i-1$
- 设指针变量 p 指向单链表结点 A，则删除结点 A 的后继结点 B 需要的操作为（.A ）。  
(A)  $p \rightarrow next = p \rightarrow next \rightarrow next$  (B)  $p = p \rightarrow next$   
(C)  $p = p \rightarrow next \rightarrow next$  (D)  $p \rightarrow next = p$
- 设栈 S 和队列 Q 的初始状态为空，元素 E1、E2、E3、E4、E5 和 E6 依次通过栈 S，一个元素出栈后即进入队列 Q，若 6 个元素出列的顺序为 E2、E4、E3、E6、E5 和 E1，则栈 S 的容量至少应该是（.C ）。  
(A) 6 (B) 4 (C) 3 (D) 2
- 将 10 阶对称矩阵压缩存储到一维数组 A 中，则数组 A 的长度最少为（C ）。  
(A) 100 (B) 40 (C) 55 (D) 80
- 设结点 A 有 3 个兄弟结点且结点 B 为结点 A 的双亲结点，则结点 B 的度数数为（8.B ）  
(A) 3 (B) 4 (C) 5 (D) 1
- 根据二叉树的定义可知二叉树共有（ B ）种不同的形态。  
(A) 4 (B) 5 (C) 6 (D) 7
- 设有以下四种排序方法，则（ B ）的空间复杂度最大。  
(A) 冒泡排序 (B) 快速排序 (C) 堆排序 (D) 希尔排序

### 二、填空题(30 分)

- 设顺序循环队列  $Q[0: m-1]$  的队头指针和队尾指针分别为 F 和 R，其中队头指针 F 指向当前队头元素的前一个位置，队尾指针 R 指向当前队尾元素所在的位置，则出队列的语句为  $F = \underline{\hspace{2cm}}$ ；。
- 设线性表中有 n 个数据元素，则在顺序存储结构上实现顺序查找的平均时间复杂度为  $\underline{\hspace{2cm}}$ ，在链式存储结构上实现顺序查找的平均时间复杂度为  $\underline{\hspace{2cm}}$ 。
- 设一棵二叉树中有 n 个结点，则当用二叉链表作为其存储结构时，该二叉链表中共有  $\underline{\hspace{2cm}}$  个指针域，  $\underline{\hspace{2cm}}$  个空指针域。
- 设指针变量 p 指向单链表中结点 A，指针变量 s 指向被插入的结点 B，则在结点 A 的后面插入结点 B 的操作序列为  $\underline{\hspace{2cm}}$ 。
- 设无向图 G 中有 n 个顶点和 e 条边，则其对应的邻接表中有  $\underline{\hspace{2cm}}$  个表头结点和  $\underline{\hspace{2cm}}$  个表结点。
- 设无向图 G 中有 n 个顶点 e 条边，所有顶点的度数之和为 m，则 e 和 m 有  $\underline{\hspace{2cm}}$  关系。
- 设一棵二叉树的前序遍历序列和中序遍历序列均为 ABC，则该二叉树的后序遍历序列为  $\underline{\hspace{2cm}}$ 。
- 设一棵完全二叉树中有 21 个结点，如果按照从上到下、从左到右的顺序从 1 开始顺序编号，则编号为 8 的双亲结点的编号是  $\underline{\hspace{2cm}}$ ，编号为 8 的左孩子结点的编号是  $\underline{\hspace{2cm}}$ 。
- 下列程序段的功能实现子串 t 在主串 s 中位置的算法，要求在下划线处填上正确语句。

```
int index(char s[ ], char t[ ])
{
    i=j=0;
    while(i<strlen(s) && j<strlen(t)) if(s[i]==t[j]){i=i+1; j=j+1;}else{i=        ; j=        ;}
    if (j==strlen(t))return(i-strlen(t));else return (-1);
}
```

10. 设一个连通图  $G$  中有  $n$  个顶点  $e$  条边, 则其最小生成树上有\_\_\_\_\_条边。

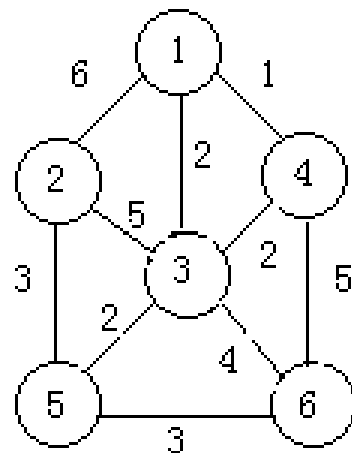
1. 设完全二叉树的顺序存储结构中存储数据 ABCDE, 要求给出该二叉树的链式存储结构并给出该二叉树的前序、中序和后序遍历序列。

2. 设给定一个权值集合  $W=(3, 5, 7, 9, 11)$ ，要求根据给定的权值集合构造一棵哈夫曼树并计算哈夫曼树的带权路径长度 WPL。

3. 设一组初始记录关键字序列为 (19, 21, 16, 5, 18, 23), 要求给出以 19 为基准的一趟快速排序结果以及第 2 趟直接选择排序后的结果。

4. 设一组初始记录关键字集合为 (25, 10, 8, 27, 32, 68)，散列表的长度为 8，散列函数  $H(k) = k \bmod 7$ ，要求分别用线性探测和链地址法作为解决冲突的方法设计哈希表。

5. 设无向图  $G$  (所右图所示), 要求给出该图的深度优先和广度优先遍历的序列并给出该图的最小生成树。



1. 设计判断单链表中结点是否关于中心对称算法。
2. 设计在链式存储结构上建立一棵二叉树的算法。
3. 设计判断一棵二叉树是否是二叉排序树的算法。

## 一、选择题

1. C      2. C      3. D      4. C      5. A   6. C      7. C      8. B      9. B      10. B

1.  $(F+1) \% m$
2.  $O(n), O(n)$
3.  $2n, n+1$

4.  $s \rightarrow next = p \rightarrow next; p \rightarrow next = s$

5. n, 2e

6.  $m=2e$

## 7. CBA

8. 4, 16

9.  $i-j+1, 0$

10.  $n-1$

1. 链式存储结构略，前序 ABDEC，中序 DBEAC，后序 DEBCA。

2. 哈夫曼树略, WPL=78

3.  $(18, 5, 16, 19, 21, 23), (5, 16, 21, 19, 18, 23)$

4. 线性探测:

0	1	2	3	4	5	6	7
$\Lambda$	8	$\Lambda$	10	25	32	27	68

$$\begin{aligned} &h_0 \\ &h_1 - > 8 \\ &h_2 \\ \text{链地址法: } &h_3 - > 10 \\ &h_4 - > 25 - > 32 \\ &h_5 - > 68 \\ &h_6 - > 27 \end{aligned}$$

5. 深度: 125364, 广度: 123456, 最小生成树 T 的边集为  $E=\{(1, 4), (1, 3), (3, 5), (5, 6), (5,6)\}$

#### 四、算法设计题

1. 设计判断单链表中结点是否关于中心对称算法。

```
typedef struct {int s[100]; int top;} sqstack;
int lklistsymmetry(lklist *head)
{
    sqstack stack;    stack.top= -1; lklist *p;
    for(p=head;p!=0;p=p->next) {stack.top++; stack.s[stack.top]=p->data;}
    for(p=head;p!=0;p=p->next) if (p->data==stack.s[stack.top]) stack.top=stack.top-1; else return(0);
    return(1);
}
```

2. 设计在链式存储结构上建立一棵二叉树的算法。

```
typedef char datatype;
typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
void createbitree(bitree *&bt)
{
    char ch; scanf("%c",&ch);
    if(ch=='#') {bt=0; return;}
    bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch;
    createbitree(bt->lchild); createbitree(bt->rchild);
}
```

3. 设计判断一棵二叉树是否是二叉排序树的算法。

```
int minnum=-32768,flag=1;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void inorder(bitree *bt)
{
    if (bt!=0)
        {inorder(bt->lchild); if(minnum>bt->key)flag=0; minnum=bt->key; inorder(bt->rchild);}
}
```

## 数据结构试卷（二）

### 一、选择题(24 分)

1. 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
2. 设哈夫曼树中的叶子结点总数为  $m$ ，若用二叉链表作为存储结构，则该哈夫曼树中总共有（ 2.B ）个空指针域。  
(A)  $2m-1$  (B)  $2m$  (C)  $2m+1$  (D)  $4m$
3. 设顺序循环队列  $Q[0:M-1]$  的头指针和尾指针分别为  $F$  和  $R$ ，头指针  $F$  总是指向队头元素的前一位置，尾指针  $R$  总是指向队尾元素的当前位置，则该循环队列中的元素个数为（ 3.C ）。  
(A)  $R-F$  (B)  $F-R$  (C)  $(R-F+M) \% M$  (D)  $(F-R+M) \% M$
4. 设某棵二叉树的中序遍历序列为 **ABCD**，前序遍历序列为 **CABD**，则后序遍历该二叉树得到序列为（ 4.A ）。  
(A) **BADC** (B) **BCDA** (C) **CDAB** (D) **CBDA**
5. 设某完全无向图中有  $n$  个顶点，则该完全无向图中有（ 5.A ）条边。  
(A)  $n(n-1)/2$  (B)  $n(n-1)$  (C)  $n^2$  (D)  $n^2-1$
6. 设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为（C）。

- (A) 9 (B) 10 (C) 11 (D) 12
7. 设某有向图中有  $n$  个顶点，则该有向图对应的邻接表中有 (B) 个表头结点。  
 (A)  $n-1$  (B)  $n$  (C)  $n+1$  (D)  $2n-1$
8. 设一组初始记录关键字序列 (5, 2, 6, 3, 8)，以第一个记录关键字 5 为基准进行一趟快速排序的结果为 (C)。  
 (A) 2, 3, 5, 8, 6 (B) 3, 2, 5, 8, 6  
 (C) 3, 2, 5, 6, 8 (D) 2, 3, 6, 5, 8

## 二、填空题(24 分)

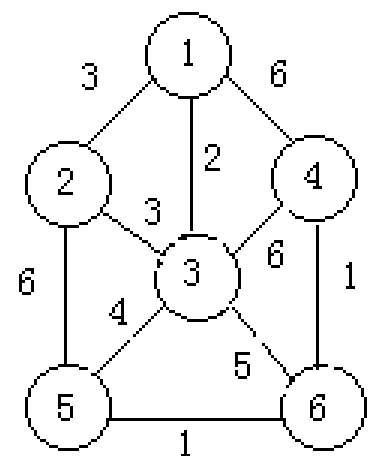
- 为了能有效地应用 HASH 查找技术，必须解决的两个问题是 构造一个好的 HASH 函数 和 确定解决冲突的方法。
- 下面程序段的功能实现数据  $x$  进栈，要求在下划线处填上正确的语句。  

```
typedef struct {int s[100]; int top;} sqstack;
void push(sqstack &stack,int x)
{
    if (stack.top==m-1) printf(“overflow”);
    else {_____};
}
```
- 中序遍历二叉排序树所得到的序列是 有序 序列 (填有序或无序)。
- 快速排序的最坏时间复杂度为  $O(n^2)$ ，平均时间复杂度为  $O(n \log n)$ 。
- 设某棵二叉树中度数为 0 的结点数为  $N_0$ ，度数为 1 的结点数为  $N_1$ ，则该二叉树中度数为 2 的结点数为  $N_0 - 1$ ；若采用二叉链表作为该二叉树的存储结构，则该二叉树中共有  $2N_0 - 2$  个空指针域。
- 设某无向图中顶点数和边数分别为  $n$  和  $e$ ，所有顶点的度数之和为  $d$ ，则  $e = \frac{d}{2}$ 。
- 设一组初始记录关键字序列为 (55, 63, 44, 38, 75, 80, 31, 56)，则利用筛选法建立的初始堆为 55。

8. 设某无向图  $G$  的邻接表为
- $$\begin{aligned} v_1 &\rightarrow 3 \rightarrow 2 \rightarrow 4 \\ v_2 &\rightarrow 1 \rightarrow 3 \\ v_3 &\rightarrow 1 \rightarrow 4 \rightarrow 2 \\ v_4 &\rightarrow 1 \rightarrow 3 \end{aligned}$$
- ，则从顶点  $V_1$  开始的深度优先遍历序列为 1, 3, 2, 4；  
 广度优先遍历序列为 1, 3, 2, 4。

## 三、应用题(36 分)

- 设一组初始记录关键字序列为 (45, 80, 48, 40, 22, 78)，则分别给出第 4 趟简单选择排序和第 4 趟直接插入排序后的结果。
- 设指针变量  $p$  指向双向链表中结点  $A$ ，指针变量  $q$  指向被插入结点  $B$ ，要求给出在结点  $A$  的后面插入结点  $B$  的操作序列 (设双向链表中结点的两个指针域分别为  $llink$  和  $rlink$ )。
- 设一组有序的记录关键字序列为 (13, 18, 24, 35, 47, 50, 62, 83, 90)，查找方法用二分查找，要求计算出查找关键字 62 时的比较次数并计算出查找成功时的平均查找长度。
- 设一棵树  $T$  中边的集合为  $\{(A, B), (A, C), (A, D), (B, E), (C, F), (C, G)\}$ ，要求用孩子兄弟表示法 (二叉链表) 表示出该树的存储结构并将该树转化成对应的二叉树。
- 设有一组初始记录关键字为 (45, 80, 48, 40, 22, 78)，要求构造一棵二叉排序树并给出构造过程。



## 四、算法设计题(16 分)

1. 设有一组初始记录关键字序列  $(K_1, K_2, \dots, K_n)$ ，要求设计一个算法能够在  $O(n)$  的时间复杂度内将线性表划分成两部分，其中左半部分的每个关键字均小于  $K_i$ ，右半部分的每个关键字均大于等于  $K_i$ 。
2. 设有两个集合 A 和集合 B，要求设计生成集合  $C=A \cap B$  的算法，其中集合 A、B 和 C 用链式存储结构表示。

## 数据结构试卷（二）参考答案

### 一、选择题

1. D      2. B      3. C      4. A      5. A      6. C      7. B      8. C

### 二、填空题

1. 构造一个好的 HASH 函数，确定解决冲突的方法
2.  $stack.top++$ ,  $stack.s[stack.top]=x$
3. 有序
4.  $O(n^2)$ ,  $O(n \log_2 n)$
5.  $N_0 - 1$ ,  $2N_0 + N_1$
6.  $d/2$
7. (31, 38, 54, 56, 75, 80, 55, 63)
8. (1, 3, 4, 2), (1, 3, 2, 4)

### 三、应用题

1. (22, 40, 45, 48, 80, 78), (40, 45, 48, 80, 22, 78)
2.  $q \rightarrow llink = p; q \rightarrow rlink = p \rightarrow rlink; p \rightarrow rlink \rightarrow llink = q; p \rightarrow rlink = q;$
3. 2,  $ASL = 91 * 1 + 2 * 2 + 3 * 4 + 4 * 2 = 25/9$
4. 树的链式存储结构略，二叉树略
5.  $E = \{(1, 3), (1, 2), (3, 5), (5, 6), (6, 4)\}$
6. 略

### 四、算法设计题

1. 设有一组初始记录关键字序列  $(K_1, K_2, \dots, K_n)$ ，要求设计一个算法能够在  $O(n)$  的时间复杂度内将线性表划分成两部分，其中左半部分的每个关键字均小于  $K_i$ ，右半部分的每个关键字均大于等于  $K_i$ 。

```
void quickpass(int r[], int s, int t)
{
    int i=s, j=t, x=r[s];
    while(i<j){
        while (i<j && r[j]>x) j=j-1; if (i<j) {r[i]=r[j];i=i+1;}
        while (i<j && r[i]<x) i=i+1; if (i<j) {r[j]=r[i];j=j-1;}
    }
    r[i]=x;
}
```

2. 设有两个集合 A 和集合 B，要求设计生成集合  $C=A \cap B$  的算法，其中集合 A、B 和 C 用链式存储结构表示。

```
typedef struct node {int data; struct node *next;} llist;
void intersection(llist *ha, llist *hb, llist *&hc)
{
    llist *p,*q,*t;
    for(p=ha, hc=0; p!=0; p=p->next)
    {   for(q=hb; q!=0; q=q->next) if (q->data==p->data) break;
        if(q!=0){ t=(llist *)malloc(sizeof(llist)); t->data=p->data; t->next=hc; hc=t;}
    }
}
```

## 数据结构试卷（三）

### 一、选择题(30 分)

1. 设某数据结构的二元组形式表示为  $A=(D, R)$ ,  $D=\{01, 02, 03, 04, 05, 06, 07, 08, 09\}$ ,  $R=\{r\}$ ,  $r=\{<01, 02>, <01, 03>, <01, 04>, <02, 05>, <02, 06>, <03, 07>, <03, 08>, <03, 09>\}$ , 则数据结构 A 是 ( B )。  
(A) 线性结构 (B) 树型结构 (C) 物理结构 (D) 图型结构
2. 下面程序的时间复杂为 ( B )  
`for (i=1, s=0; i<=n; i++) {t=1; for(j=1; j<=i; j++) t=t*j; s=s+t; }`  
(A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n^3)$  (D)  $O(n^4)$
3. 设指针变量 p 指向单链表中结点 A, 若删除单链表中结点 A, 则需要修改指针的操作序列为 ( A )。  
(A)  $q=p \rightarrow next$ ;  $p \rightarrow data=q \rightarrow data$ ;  $p \rightarrow next=q \rightarrow next$ ;  $free(q)$ ;  
(B)  $q=p \rightarrow next$ ;  $q \rightarrow data=p \rightarrow data$ ;  $p \rightarrow next=q \rightarrow next$ ;  $free(q)$ ;  
(C)  $q=p \rightarrow next$ ;  $p \rightarrow next=q \rightarrow next$ ;  $free(q)$ ;  
(D)  $q=p \rightarrow next$ ;  $p \rightarrow data=q \rightarrow data$ ;  $free(q)$ ;
4. 设有 n 个待排序的记录关键字, 则在堆排序中需要 ( A ) 个辅助记录单元。  
(A) 1 (B) n (C)  $n \log_2 n$  (D)  $n^2$
5. 设一组初始关键字记录关键字为 (20, 15, 14, 18, 21, 36, 40, 10), 则以 20 为基准记录的一趟快速排序结束后的结果为 ( A )。  
(A) 10, 15, 14, 18, 20, 36, 40, 21  
(B) 10, 15, 14, 18, 20, 40, 36, 21  
(C) 10, 15, 14, 20, 18, 40, 36, 21  
(D) 15, 10, 14, 18, 20, 36, 40, 21
6. 设二叉排序树中有 n 个结点, 则在二叉排序树的平均平均查找长度为 ( B )。  
(A)  $O(1)$  (B)  $O(\log_2 n)$  (C)  $O(n)$  (D)  $O(n^2)$
7. 设无向图 G 中有 n 个顶点 e 条边, 则其对应的邻接表中的表头结点和表结点的个数分别为 ( D )。  
(A) n, e (B) e, n (C) 2n, e (D) n, 2e
8. 设某强连通图中有 n 个顶点, 则该强连通图中至少有 ( C ) 条边。  
(A)  $n(n-1)$  (B)  $n+1$  (C) n (D)  $n(n+1)$
9. 设有 5000 个待排序的记录关键字, 如果需要用最快的方法选出其中最小的 10 个记录关键字, 则用下列 ( B ) 方法可以达到此目的。  
(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 插入排序
10. 下列四种排序中 ( D ) 的空间复杂度最大。  
(A) 插入排序 (B) 冒泡排序 (C) 堆排序 (D) 归并排序

### 二、填空题(48 分, 其中最后两小题各 6 分)

1. 数据的物理结构主要包括\_\_\_\_\_和\_\_\_\_\_两种情况。
2. 设一棵完全二叉树中有 500 个结点, 则该二叉树的深度为\_\_\_\_9\_\_\_\_; 若用二叉链表作为该完全二叉树的存储结构, 则共有\_\_\_\_501\_\_\_\_个空指针域。
3. 设输入序列为 1、2、3, 则经过栈的作用后可以得到\_\_\_\_5\_\_\_\_种不同的输出序列。
4. 设有向图 G 用邻接矩阵  $A[n][n]$  作为存储结构, 则该邻接矩阵中第 i 行上所有元素之和等于顶点 i 的\_\_\_\_出度\_\_\_\_, 第 i 列上所有元素之和等于顶点 i 的\_\_\_\_入度\_\_\_\_。
5. 设哈夫曼树中共有 n 个结点, 则该哈夫曼树中有\_\_\_\_0\_\_\_\_个度数为 1 的结点。
6. 设有向图 G 中有 n 个顶点 e 条有向边, 所有的顶点入度数之和为 d, 则 e 和 d 的关系为\_\_\_\_ $e=d$ \_\_\_\_。
7. \_\_\_\_中序\_\_\_\_遍历二叉排序树中的结点可以得到一个递增的关键字序列 (填先序、中序或后序)。
8. 设查找表中有 100 个元素, 如果用二分法查找方法查找数据元素 X, 则最多需要比较\_\_\_\_7\_\_\_\_次就可以断定数据元素 X 是否在查找表中。

9. 不论是顺序存储结构的栈还是链式存储结构的栈，其入栈和出栈操作的时间复杂度均为  $O(n)$ 。
10. 设有  $n$  个结点的完全二叉树，如果按照从自上到下、从左到右从 1 开始顺序编号，则第  $i$  个结点的双亲结点编号为  $\lfloor i/2 \rfloor$ ，右孩子结点的编号为  $2i+1$ 。
11. 设一组初始记录关键字为 (72, 73, 71, 23, 94, 16, 5)，则以记录关键字 72 为基准的一趟快速排序结果为 (5, 16, 23, 71, 72, 73, 94)。
12. 设有向图  $G$  中有向边的集合  $E=\{<1, 2>, <2, 3>, <1, 4>, <4, 2>, <4, 3>\}$ ，则该图的一种拓扑序列为 1, 4, 2, 3。
13. 下列算法实现在顺序散列表中查找值为  $x$  的关键字，请在下划线处填上正确的语句。

```
struct record{int key; int others;};
int hashsqsearch(struct record hashtable[ ],int k)
{
    int i,j;   j=i=k % p;
    while (hashtable[j].key!=k&&hashtable[j].flag!=0){j=(j+1) % m; if (i==j) return(-1);}
    if (_____ ) return(j); else return(-1);
}
```

14. 下列算法实现在二叉排序树上查找关键值  $k$ ，请在下划线处填上正确的语句。

```
typedef struct node{int key; struct node *lchild; struct node *rchild;}bitree;
bitree *bstsearch(bitree *t, int k)
{
    if (t==0) return(0);else while (t!=0)
        if (t->key==k)_____; else if (t->key>k) t=t->lchild; else _____;
}
```

### 三、算法设计题 (22 分)

1. 设计在单链表中删除值相同的多余结点的算法。
2. 设计一个求结点  $x$  在二叉树中的双亲结点算法。

## 数据结构试卷（三）参考答案

### 一、选择题

1. B      2. B      3. A      4. A      5. A  
6. B      7. D      8. C      9. B      10. D

第 3 小题分析：首先用指针变量  $q$  指向结点  $A$  的后继结点  $B$ ，然后将结点  $B$  的值复制到结点  $A$  中，最后删除结点  $B$ 。

第 9 小题分析：9 快速排序、归并排序和插入排序必须等到整个排序结束后才能够求出最小的 10 个数，而堆排序只需要在初始堆的基础上再进行 10 次筛选即可，每次筛选的时间复杂度为  $O(\log_2 n)$ 。

### 二、填空题

1. 顺序存储结构、链式存储结构
2. 9, 501
3. 5
4. 出度，入度
5. 0
6.  $e=d$
7. 中序
8. 7
9.  $O(1)$
10.  $i/2$ ,  $2i+1$
11. (5, 16, 71, 23, 72, 94, 73)

12. (1, 4, 3, 2)

13. `j+1, hashtable[j].key==k`

14. `return(t), t=t->rchild`

第 8 小题分析：二分查找的过程可以用一棵二叉树来描述，该二叉树称为二叉判定树。在有序表上进行二分查找时的查找长度不超过二叉判定树的高度  $1+\log_2 n$ 。

### 三、算法设计题

1. 设计在单链表中删除值相同的多余结点的算法。

```
typedef int datatype;
typedef struct node {datatype data; struct node *next;} llist;
void delredundant(llist *&head)
{
    llist *p,*q,*s;
    for(p=head;p!=0;p=p->next)
    {
        for(q=p->next,s=q;q!=0;)
            if (q->data==p->data) {s->next=q->next; free(q);q=s->next;}
            else {s=q,q=q->next;}
    }
}
```

2. 设计一个求结点  $x$  在二叉树中的双亲结点算法。

```
typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
bitree *q[20]; int r=0,f=0,flag=0;
void preorder(bitree *bt, char x)
{
    if (bt!=0 && flag==0)
        if (bt->data==x) { flag=1; return;}
        else {r=(r+1)% 20; q[r]=bt; preorder(bt->lchild,x); preorder(bt->rchild,x); }
}
void parent(bitree *bt,char x)
{
    int i;
    preorder(bt,x);
    for(i=f+1; i<=r; i++) if (q[i]->lchild->data==x || q[i]->rchild->data==x) break;
    if (flag==0) printf("not found x\n");
    else if (i<=r) printf("%c",bt->data); else printf("not parent");
}
```

### 数据结构试卷（四）

#### 一、选择题(30 分)

1. 设一维数组中有  $n$  个数组元素，则读取第  $i$  个数组元素的平均时间复杂度为 ( )。

(A)  $O(n)$  (B)  $O(n\log_2 n)$  (C)  $O(1)$  (D)  $O(n^2)$

2. 设一棵二叉树的深度为  $k$ ，则该二叉树中最多有 ( ) 个结点。

(A)  $2k-1$  (B)  $2^k$  (C)  $2^{k-1}$  (D)  $2^{k-1}-1$

3. 设某无向图中有  $n$  个顶点  $e$  条边，则该无向图中所有顶点的入度之和为 ( )。

(A)  $n$  (B)  $e$  (C)  $2n$  (D)  $2e$

4. 在二叉排序树中插入一个结点的时间复杂度为 ( )。

(A)  $O(1)$  (B)  $O(n)$  (C)  $O(\log_2 n)$  (D)  $O(n^2)$



5. 设某有向图的邻接表中有  $n$  个表头结点和  $m$  个表结点, 则该图中有 ( ) 条有向边。  
(A)  $n$  (B)  $n-1$  (C)  $m$  (D)  $m-1$
6. 设一组初始记录关键字序列为 (345, 253, 674, 924, 627), 则用基数排序需要进行 ( ) 趟的分配和回收才能使得初始关键字序列变成有序序列。  
(A) 3 (B) 4 (C) 5 (D) 8
7. 设用链表作为栈的存储结构则退栈操作 ( )。  
(A) 必须判别栈是否为满 (B) 必须判别栈是否为空  
(C) 判别栈元素的类型 (D) 对栈不作任何判别
8. 下列四种排序中 ( ) 的空间复杂度最大。  
(A) 快速排序 (B) 冒泡排序 (C) 希尔排序 (D) 堆
9. 设某二叉树中度数为 0 的结点数为  $N_0$ , 度数为 1 的结点数为  $N_1$ , 度数为 2 的结点数为  $N_2$ , 则下列等式成立的是 ( )。  
(A)  $N_0 = N_1 + 1$  (B)  $N_0 = N_1 + N_2$  (C)  $N_0 = N_2 + 1$  (D)  $N_0 = 2N_1 + 1$
10. 设有序顺序表中有  $n$  个数据元素, 则利用二分查找法查找数据元素  $X$  的最多比较次数不超过 ( )。  
(A)  $\log_2 n + 1$  (B)  $\log_2 n - 1$  (C)  $\log_2 n$  (D)  $\log_2 (n+1)$

## 二、填空题(42 分)

1. 设有  $n$  个无序的记录关键字, 则直接插入排序的时间复杂度为\_\_\_\_\_, 快速排序的平均时间复杂度为\_\_\_\_\_。
2. 设指针变量  $p$  指向双向循环链表中的结点  $X$ , 则删除结点  $X$  需要执行的语句序列为\_\_\_\_\_ (设结点中的两个指针域分别为  $llink$  和  $rlink$ )。
3. 根据初始关键字序列 (19, 22, 01, 38, 10) 建立的二叉排序树的高度为\_\_\_\_\_。
4. 深度为  $k$  的完全二叉树中最少有\_\_\_\_\_个结点。
5. 设初始记录关键字序列为  $(K_1, K_2, \dots, K_n)$ , 则用筛选法思想建堆必须从第\_\_\_\_\_个元素开始进行筛选。
6. 设哈夫曼树中共有 99 个结点, 则该树中有\_\_\_\_\_个叶子结点; 若采用二叉链表作为存储结构, 则该树中有\_\_\_\_\_个空指针域。
7. 设有一个顺序循环队列中有  $M$  个存储单元, 则该循环队列中最多能够存储\_\_\_\_\_个队列元素; 当前实际存储\_\_\_\_\_个队列元素 (设头指针  $F$  指向当前队头元素的前一个位置, 尾指针指向当前队尾元素的位置)。
8. 设顺序线性表中有  $n$  个数据元素, 则第  $i$  个位置上插入一个数据元素需要移动表中\_\_\_\_\_个数据元素; 删除第  $i$  个位置上的数据元素需要移动表中\_\_\_\_\_个元素。
9. 设一组初始记录关键字序列为 (20, 18, 22, 16, 30, 19), 则以 20 为中轴的一趟快速排序结果为\_\_\_\_\_。
10. 设一组初始记录关键字序列为 (20, 18, 22, 16, 30, 19), 则根据这些初始关键字序列建成的初始堆为\_\_\_\_\_。
11. 设某无向图  $G$  中有  $n$  个顶点, 用邻接矩阵  $A$  作为该图的存储结构, 则顶点  $i$  和顶点  $j$  互为邻接点的条件是\_\_\_\_\_。
12. 设无向图对应的邻接矩阵为  $A$ , 则  $A$  中第  $i$  上非 0 元素的个数\_\_\_\_\_第  $i$  列上非 0 元素的个数 (填等于, 大于或小于)。
13. 设前序遍历某二叉树的序列为  $ABCD$ , 中序遍历该二叉树的序列为  $BADC$ , 则后序遍历该二叉树的序列为\_\_\_\_\_。
14. 设散列函数  $H(k)=k \bmod p$ , 解决冲突的方法为链地址法。要求在下列算法划线处填上正确的语句完成在散列表 `hashtalbe` 中查找关键字值等于  $k$  的结点, 成功时返回指向关键字的指针, 不成功时返回标志 0。

```
typedef struct node {int key; struct node *next;} lklist;
void createlkhash(lklist *hashtable[ ])
```

```

{
    int i,k;   lklist *s;
    for(i=0;i<m;i++)_____
    for(i=0;i<n;i++)
    {
        s=(lklist *)malloc(sizeof(lklist)); s->key=a[i];
        k=a[i] % p; s->next=hashtable[k];_____
    }
}

```

### 三、算法设计题(28 分)

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。
2. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。
3. 在链式存储结构上建立一棵二叉排序树。

## 数据结构试卷（四）参考答案

### 一、选择题

1. C    2. D    3. D    4. B    5. C    6. A    7. B    8. A    9. C    10. A

### 二、填空题

1.  $O(n^2)$ ,  $O(n \log_2 n)$
2.  $p \rightarrow llink \rightarrow rlink = p \rightarrow rlink$ ;  $p \rightarrow rlink \rightarrow llink = p \rightarrow llink$
3. 3
4.  $2^{k-1}$
5.  $n/2$
6. 50, 51
7.  $m-1$ ,  $(R-F+M)\%M$
8.  $n+1-i$ ,  $n-i$
9. (19, 18, 16, 20, 30, 22)
10. (16, 18, 19, 20, 32, 22)
11.  $A[i][j]=1$
12. 等于
13. BDCA
14.  $hashtable[i]=0$ ,  $hashtable[k]=s$

### 三、算法设计题

1. 设单链表中有仅三类字符的数据元素(大写字母、数字和其它字符)，要求利用原单链表中结点空间设计出三个单链表的算法，使每个单链表只包含同类字符。

```

typedef char datatype;
typedef struct node {datatype data; struct node *next;}lklist;
void split(lklist *head,lklist *&ha,lklist *&hb,lklist *&hc)
{
    lklist *p; ha=0,hb=0,hc=0;
    for(p=head;p!=0;p=head)
    {
        head=p->next; p->next=0;
        if (p->data>='A' && p->data<='Z') {p->next=ha; ha=p;}
        else if (p->data>='0' && p->data<='9') {p->next=hb; hb=p;} else {p->next=hc; hc=p;}
    }
}

```

- ```

    }
2. 设计在链式存储结构上交换二叉树中所有结点左右子树的算法。
    typedef struct node {int data; struct node *lchild,*rchild;} bitree;
    void swapbitree(bitree *bt)
    {
        bitree *p;
        if(bt==0) return;
        swapbitree(bt->lchild); swapbitree(bt->rchild);
        p=bt->lchild; bt->lchild=bt->rchild; bt->rchild=p;
    }
3. 在链式存储结构上建立一棵二叉排序树。
    #define n 10
    typedef struct node{int key; struct node *lchild,*rchild;}bitree;
    void bstinsert(bitree *&bt,int key)
    {
        if (bt==0){bt=(bitree *)malloc(sizeof(bitree)); bt->key=key;bt->lchild=bt->rchild=0;}
        else if (bt->key>key) bstinsert(bt->lchild,key); else bstinsert(bt->rchild,key);
    }
    void createbsttree(bitree *&bt)
    {
        int i;
        for(i=1;i<=n;i++) bstinsert(bt,random(100));
    }

```

## 数据结构试卷（五）

### 一、选择题(30 分)

- 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
- 设一组初始记录关键字序列为(50, 40, 95, 20, 15, 70, 60, 45)，则以增量 d=4 的一趟希尔排序结束后前 4 条记录关键字为（ ）。  
(A) 40, 50, 20, 95 (B) 15, 40, 60, 20  
(C) 15, 20, 40, 45 (D) 45, 40, 15, 20
- 设一组初始记录关键字序列为(25, 50, 15, 35, 80, 85, 20, 40, 36, 70)，其中含有 5 个长度为 2 的有序子表，则用归并排序的方法对该记录关键字序列进行一趟归并后的结果为（ ）。  
(A) 15, 25, 35, 50, 20, 40, 80, 85, 36, 70  
(B) 15, 25, 35, 50, 80, 20, 85, 40, 70, 36  
(C) 15, 25, 35, 50, 80, 85, 20, 36, 40, 70  
(D) 15, 25, 35, 50, 80, 20, 36, 40, 70, 85
- 函数 substr(“DATASTRUCTURE”, 5, 9)的返回值为（ ）。  
(A) “STRUCTURE” (B) “DATA”  
(C) “ASTRUCTUR” (D) “DATASTRUCTURE”
- 设一个有序的单链表中有 n 个结点，现要求插入一个新结点后使得单链表仍然保持有序，则该操作的时间复杂度为（ ）。  
(A)  $O(\log_2 n)$  (B)  $O(1)$  (C)  $O(n^2)$  (D)  $O(n)$
- 设一棵 m 叉树中度数为 0 的结点数为  $N_0$ ，度数为 1 的结点数为  $N_1$ ，……，度数为 m 的结点数为  $N_m$ ，则  $N =$ （ ）。  
(A)  $N_1 + N_2 + \dots + N_m$  (B)  $1 + N_2 + 2N_3 + 3N_4 + \dots + (m-1)N_m$

- (C)  $N_2 + 2N_3 + 3N_4 + \dots + (m-1)N_m$  (D)  $2N_1 + 3N_2 + \dots + (m+1)N_m$
7. 设有序表中有 1000 个元素，则用二分查找查找元素 X 最多需要比较 ( ) 次。  
(A) 25 (B) 10 (C) 7 (D) 1
8. 设连通图 G 中的边集  $E = \{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点 a 出发可以得到一种深度优先遍历的顶点序列为 ( )。  
(A) abedfc (B) acfebd (C) aebdfc (D) aedfcb
9. 设输入序列是 1、2、3、……、n，经过栈的作用后输出序列的第一个元素是 n，则输出序列中第 i 个输出元素是 ( )。  
(A) n-i (B) n-1-i (C) n+1-i (D) 不能确定
- 10 设一组初始记录关键字序列为 (45, 80, 55, 40, 42, 85)，则以第一个记录关键字 45 为基准而得到一趟快速排序的结果是 ( C )。  
(A) 40, 42, 45, 55, 80, 83 (B) 42, 40, 45, 80, 85, 88  
(C) 42, 40, 45, 55, 80, 85 (D) 42, 40, 45, 85, 55, 80

## 二、填空题(共 30 分)

1. 设有一个顺序共享栈  $S[0:n-1]$ ，其中第一个栈项指针  $top1$  的初值为 -1，第二个栈项指针  $top2$  的初值为 n，则判断共享栈满的条件是\_\_\_\_\_。
2. 在图的邻接表中用顺序存储结构存储表头结点的优点是\_\_\_\_\_。
3. 设有一个 n 阶的下三角矩阵 A，如果按照行的顺序将下三角矩阵中的元素（包括对角线上元素）存放在  $n(n+1)$  个连续的存储单元中，则  $A[i][j]$  与  $A[0][0]$  之间有\_\_\_\_\_个数据元素。
4. 栈的插入和删除只能在栈的栈顶进行，后进栈的元素必定先出栈，所以又把栈称为\_\_\_\_\_表；队列的插入和删除运算分别在队列的两端进行，先进队列的元素必定先出队列，所以又把队列称为\_\_\_\_\_表。
5. 设一棵完全二叉树的顺序存储结构中存储数据元素为 ABCDEF，则该二叉树的前序遍历序列为\_\_\_\_\_，中序遍历序列为\_\_\_\_\_，后序遍历序列为\_\_\_\_\_。
6. 设一棵完全二叉树有 128 个结点，则该完全二叉树的深度为\_\_\_\_\_，有\_\_\_\_\_个叶子结点。
7. 设有向图 G 的存储结构用邻接矩阵 A 来表示，则 A 中第 i 行中所有非零元素个数之和等于顶点 i 的\_\_\_\_\_，第 i 列中所有非零元素个数之和等于顶点 i 的\_\_\_\_\_。
8. 设一组初始记录关键字序列  $(k_1, k_2, \dots, k_n)$  是堆，则对  $i=1, 2, \dots, n/2$  而言满足的条件为\_\_\_\_\_。
9. 下面程序段的功能是实现冒泡排序算法，请在下划线处填上正确的语句。

```
void bubble(int r[n])
{
    for(i=1; i<=n-1; i++)
    {
        for(exchange=0, j=0; j<_____; j++)
            if (r[j]>r[j+1]){temp=r[j+1];_____;r[j]=temp;exchange=1;}
        if (exchange==0) return;
    }
}
```

10. 下面程序段的功能是实现二分查找算法，请在下划线处填上正确的语句。

```
struct record{int key; int others;};
int bisearch(struct record r[ ], int k)
{
    int low=0, mid, high=n-1;
    while(low<=high)
    {
```

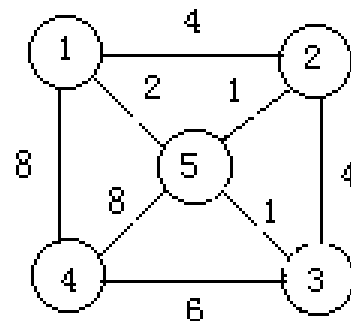
```

        _____;
        if(r[mid].key==k) return(mid+1); else if(_____) high=mid-1;else low=mid+1;
    }
    return(0);
}

```

### 三、应用题 (24 分)

1. 设某棵二叉树的中序遍历序列为 DBEAC，前序遍历序列为 ABDEC，要求给出该二叉树的后序遍历序列。
2. 设无向图 G（如右图所示），给出该图的最小生成树上边的集合并计算最小生成树各边上的权值之和。
3. 设一组初始记录关键字序列为 (15, 17, 18, 22, 35, 51, 60)，要求计算出成功查找时的平均查找长度。
4. 设散列表的长度为 8，散列函数  $H(k)=k \bmod 7$ ，初始记录关键字序列为 (25, 31, 8, 27, 13, 68)，要求分别计算出用线性探测法和链地址法作为解决冲突方法的平均查找长度。



### 四、算法设计题 (16 分)

1. 设计判断两个二叉树是否相同的算法。
2. 设计两个有序单链表的合并排序算法。

## 数据结构试卷（五）参考答案

### 一、选择题

1. A 2. B 3. A 4. A 5. D 6. B 7. B 8. B 9. C 10. C

### 二、填空题

1.  $top1+1=top2$
2. 可以随机访问到任一个顶点的简单链表
3.  $i(i+1)/2+j-1$
4. FILO, FIFO
5. ABDECF, DBEAFC, DEBFCA
6. 8, 64
7. 出度, 入度
8.  $k_i \leq k_{2i} \ \&\& \ k_i \leq k_{2i+1}$
9.  $n-i, r[j+1]=r[j]$
10.  $mid=(low+high)/2, r[mid].key>k$

### 三、应用题

1. DEBCA
2.  $E=\{(1,5),(5,2),(5,3),(3,4)\}, W=10$
3.  $ASL=(1*1+2*2+3*4)/7=17/7$
4.  $ASL1=7/6, ASL2=4/3$

### 四、算法设计题

1. 设计判断两个二叉树是否相同的算法。

```

typedef struct node {datatype data; struct node *lchild,*rchild;} bitree;
int judgebitree(bitree *bt1,bitree *bt2)
{
    if (bt1==0 && bt2==0) return(1);
    else if (bt1==0 || bt2==0 || bt1->data!=bt2->data) return(0);
    else return(judgebitree(bt1->lchild, bt2->lchild)*judgebitree(bt1->rchild, bt2->rchild));
}

```

2. 设计两个有序单链表的合并排序算法。

```
void mergelklist(lklist *ha, lklist *hb, lklist *&hc)
{
    lklist *s=hc=0;
    while(ha!=0 && hb!=0)
        if(ha->data<hb->data){ if(s==0) hc=s=ha; else {s->next=ha; s=ha;};ha=ha->next;}
        else {if(s==0) hc=s=hb; else {s->next=hb; s=hb;};hb=hb->next;}
    if(ha==0) s->next=hb; else s->next=ha;
}
```

## 数据结构试卷（六）

### 一、选择题(30 分)

1. 设一组权值集合  $W=\{2, 3, 4, 5, 6\}$ ，则由该权值集合构造的哈夫曼树中带权路径长度之和为（ ）。  
(A) 20 (B) 30 (C) 40 (D) 45
2. 执行一趟快速排序能够得到的序列是（ ）。  
(A) [41, 12, 34, 45, 27] 55 [72, 63]  
(B) [45, 34, 12, 41] 55 [72, 63, 27]  
(C) [63, 12, 34, 45, 27] 55 [41, 72]  
(D) [12, 27, 45, 41] 55 [34, 63, 72]
3. 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) head==0 (B) head->next==0  
(C) head->next==head (D) head!=0
4. 时间复杂度不受数据初始状态影响而恒为  $O(n\log_2 n)$  的是（ ）。  
(A) 堆排序 (B) 冒泡排序 (C) 希尔排序 (D) 快速排序
5. 设二叉树的先序遍历序列和后序遍历序列正好相反，则该二叉树满足的条件是（ ）。  
(A) 空或只有一个结点 (B) 高度等于其结点数  
(C) 任一结点无左孩子 (D) 任一结点无右孩子
6. 一趟排序结束后不一定能够选出一个元素放在其最终位置上的的是（ ）。  
(A) 堆排序 (B) 冒泡排序 (C) 快速排序 (D) 希尔排序
7. 设某棵三叉树中有 40 个结点，则该三叉树的最小高度为（ ）。  
(A) 3 (B) 4 (C) 5 (D) 6
8. 顺序查找不论在顺序线性表中还是在链式线性表中的时间复杂度为（ ）。  
(A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n^{1/2})$  (D)  $O(\log_2 n)$
9. 二路归并排序的时间复杂度为（ ）。  
(A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n\log_2 n)$  (D)  $O(\log_2 n)$
10. 深度为 k 的完全二叉树中最少有（ ）个结点。  
(A)  $2^{k-1}-1$  (B)  $2^{k-1}$  (C)  $2^{k-1}+1$  (D)  $2^k-1$
11. 设指针变量 front 表示链式队列的队头指针，指针变量 rear 表示链式队列的队尾指针，指针变量 s 指向将要入队列的结点 X，则入队列的操作序列为（ ）。  
(A) front->next=s; front=s; (B) s->next=rear; rear=s;  
(C) rear->next=s; rear=s; (D) s->next=front; front=s;
12. 设某无向图中有 n 个顶点 e 条边，则建立该图邻接表的时间复杂度为（ ）。  
(A)  $O(n+e)$  (B)  $O(n^2)$  (C)  $O(ne)$  (D)  $O(n^3)$
13. 设某哈夫曼树中有 199 个结点，则该哈夫曼树中有（ ）个叶子结点。  
(A) 99 (B) 100 (C) 101 (D) 102
14. 设二叉排序树上有 n 个结点，则在二叉排序树上查找结点的平均时间复杂度为（ ）。  
(A)  $O(n)$  (B)  $O(n^2)$  (C)  $O(n\log_2 n)$  (D)  $O(\log_2 n)$

15. 设用邻接矩阵 A 表示有向图 G 的存储结构, 则有向图 G 中顶点 i 的入度为 ( )。
- (A) 第 i 行非 0 元素的个数之和      (B) 第 i 列非 0 元素的个数之和
- (C) 第 i 行 0 元素的个数之和      (D) 第 i 列 0 元素的个数之和

## 二、判断题(20 分)

1. 调用一次深度优先遍历可以访问到图中的所有顶点。( )
2. 分块查找的平均查找长度不仅与索引表的长度有关, 而且与块的长度有关。( )
3. 冒泡排序在初始关键字序列为逆序的情况下执行的交换次数最多。( )
4. 满二叉树一定是完全二叉树, 完全二叉树不一定是满二叉树。( )
5. 设一棵二叉树的先序序列和后序序列, 则能够唯一确定出该二叉树的形状。( )
6. 层次遍历初始堆可以得到一个有序的序列。( )
7. 设一棵树 T 可以转化成二叉树 BT, 则二叉树 BT 中一定没有右子树。( )
8. 线性表的顺序存储结构比链式存储结构更好。( )
9. 中序遍历二叉排序树可以得到一个有序的序列。( )
10. 快速排序是排序算法中平均性能最好的一种排序。( )

## 三、填空题(30 分)

1. `for(i=1, t=1, s=0; i<=n; i++) {t=t*i; s=s+t; }` 的时间复杂度为\_\_\_\_\_。
2. 设指针变量 p 指向单链表中结点 A, 指针变量 s 指向被插入的新结点 X, 则进行插入操作的语句序列为\_\_\_\_\_ (设结点的指针域为 next)。
3. 设有向图 G 的二元组形式表示为  $G = (D, R)$ ,  $D = \{1, 2, 3, 4, 5\}$ ,  $R = \{r\}$ ,  $r = \{<1,2>, <2,4>, <4,5>, <1,3>, <3,2>, <3,5>\}$ , 则给出该图的一种拓扑排序序列\_\_\_\_\_。
4. 设无向图 G 中有 n 个顶点, 则该无向图中每个顶点的度数最多是\_\_\_\_\_。
5. 设二叉树中度数为 0 的结点数为 50, 度数为 1 的结点数为 30, 则该二叉树中总共有\_\_\_\_\_个结点数。
6. 设 F 和 R 分别表示顺序循环队列的头指针和尾指针, 则判断该循环队列为空的条件为\_\_\_\_\_。
7. 设二叉树中结点的两个指针域分别为 lchild 和 rchild, 则判断指针变量 p 所指向的结点为叶子结点的条件是\_\_\_\_\_。
8. 简单选择排序和直接插入排序算法的平均时间复杂度为\_\_\_\_\_。
9. 快速排序算法的空间复杂度平均情况下为\_\_\_\_\_, 最坏的情况下为\_\_\_\_\_。
10. 散列表中解决冲突的两种方法是\_\_\_\_\_和\_\_\_\_\_。

## 四、算法设计题(20 分)

1. 设计在顺序有序表中实现二分查找的算法。
2. 设计判断二叉树是否为二叉排序树的算法。
3. 在链式存储结构上设计直接插入排序算法

# 数据结构试卷(六) 参考答案

## 一、选择题

1. D      2. A      3. A      4. A      5. D
6. D      7. B      8. A      9. C      10. B
11. C    12. A    13. B    14. D    15. B

## 二、判断题

1. 错    2. 对    3. 对    4. 对    5. 错
6. 错    7. 对    8. 错    9. 对    10. 对

## 三、填空题

1.  $O(n)$
2. `s->next=p->next; p->next=s`
3. (1, 3, 2, 4, 5)

4.  $n-1$
5. 129
6.  $F==R$
7.  $p->lchild==0 \& \& p->rchild==0$
8.  $O(n^2)$
9.  $O(n \log_2 n)$ ,  $O(n)$
10. 开放定址法, 链地址法

#### 四、算法设计题

1. 设计在顺序有序表中实现二分查找的算法。

```
struct record {int key; int others;};
int biseach(struct record r[ ], int k)
{
    int low=0,mid,high=n-1;
    while(low<=high)
    {
        mid=(low+high)/2;
        if(r[mid].key==k) return(mid+1); else if(r[mid].key>k) high=mid-1; else low=mid+1;
    }
    return(0);
}
```

2. 设计判断二叉树是否为二叉排序树的算法。

```
int minnum=-32768,flag=1;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void inorder(bitree *bt)
{
    if (bt!=0) {inorder(bt->lchild); if(minnum>bt->key)flag=0; minnum=bt->key;inorder(bt->rchild);}
}
```

3. 在链式存储结构上设计直接插入排序算法

```
void straightinsertsort(lklist *&head)
{
    lklist *s,*p,*q; int t;
    if (head==0 || head->next==0) return;
    else for(q=head,p=head->next;p!=0;p=q->next)
    {
        for(s=head;s!=q->next;s=s->next) if (s->data>p->data) break;
        if(s==q->next)q=p;
        else{q->next=p->next; p->next=s->next; s->next=p; t=p->data;p->data=s->data;s->data=t;}
    }
}
```

#### 数据结构试卷（七）

##### 一、选择题 (30 分)

1. 设某无向图有  $n$  个顶点, 则该无向图的邻接表中有 ( ) 个表头结点。  
 (A)  $2n$  (B)  $n$  (C)  $n/2$  (D)  $n(n-1)$
2. 设无向图  $G$  中有  $n$  个顶点, 则该无向图的最小生成树上有 ( ) 条边。  
 (A)  $n$  (B)  $n-1$  (C)  $2n$  (D)  $2n-1$



3. 设一组初始记录关键字序列为 (60, 80, 55, 40, 42, 85), 则以第一个关键字 45 为基准而得到的一趟快速排序结果是 ( )。  
 (A) 40, 42, 60, 55, 80, 85 (B) 42, 45, 55, 60, 85, 80  
 (C) 42, 40, 55, 60, 80, 85 (D) 42, 40, 60, 85, 55, 80
4. ( ) 二叉排序树可以得到一个从小到大的有序序列。  
 (A) 先序遍历 (B) 中序遍历 (C) 后序遍历 (D) 层次遍历
5. 设按照从上到下、从左到右的顺序从 1 开始对完全二叉树进行顺序编号, 则编号为  $i$  结点的左孩子结点的编号为 ( )。  
 (A)  $2i+1$  (B)  $2i$  (C)  $i/2$  (D)  $2i-1$
6. 程序段  $s=i=0$ ;  $\text{do } \{i=i+1; \quad s=s+i; \} \text{while}(i \leq n)$ ; 的时间复杂度为 ( )。  
 (A)  $O(n)$  (B)  $O(n \log_2 n)$  (C)  $O(n^2)$  (D)  $O(n^3/2)$
7. 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 ( )。  
 (A)  $\text{head}==0$  (B)  $\text{head} \rightarrow \text{next}==0$   
 (C)  $\text{head} \rightarrow \text{next}==\text{head}$  (D)  $\text{head}!=0$
8. 设某棵二叉树的高度为 10, 则该二叉树上叶子结点最多有 ( )。  
 (A) 20 (B) 256 (C) 512 (D) 1024
9. 设一组初始记录关键字序列为 (13, 18, 24, 35, 47, 50, 62, 83, 90, 115, 134), 则利用二分法查找关键字 90 需要比较的关键字个数为 ( )。  
 (A) 1 (B) 2 (C) 3 (D) 4
10. 设指针变量 top 指向当前链式栈的栈顶, 则删除栈顶元素的操作序列为 ( )。  
 (A)  $\text{top}=\text{top}+1$ ; (B)  $\text{top}=\text{top}-1$ ;  
 (C)  $\text{top} \rightarrow \text{next}=\text{top}$ ; (D)  $\text{top}=\text{top} \rightarrow \text{next}$ ;

## 二、判断题(20 分)

1. 不论是入队列操作还是入栈操作, 在顺序存储结构上都需要考虑“溢出”情况。( )
2. 当向二叉排序树中插入一个结点, 则该结点一定成为叶子结点。( )
3. 设某堆中有  $n$  个结点, 则在该堆中插入一个新结点的时间复杂度为  $O(\log_2 n)$ 。( )
4. 完全二叉树中的叶子结点只可能在最后两层中出现。( )
5. 哈夫曼树中没有度数为 1 的结点。( )
6. 对连通图进行深度优先遍历可以访问到该图中的所有顶点。( )
7. 先序遍历一棵二叉排序树得到的结点序列不一定是有序的序列。( )
8. 由树转化成二叉树, 该二叉树的右子树不一定为空。( )
9. 线性表中的所有元素都有一个前驱元素和后继元素。( )
10. 带权无向图的最小生成树是唯一的。( )

## 三、填空题(30 分)

1. 设指针变量 p 指向双向链表中的结点 A, 指针变量 s 指向被插入的结点 X, 则在结点 A 的后面插入结点 X 的操作序列为 \_\_\_\_\_= $p$ ;  $s \rightarrow \text{right}=p \rightarrow \text{right}$ ; \_\_\_\_\_= $s$ ;  $p \rightarrow \text{right} \rightarrow \text{left}=s$ ; (设结点中的两个指针域分别为 left 和 right)。
2. 设完全有向图中有  $n$  个顶点, 则该完全有向图中共有 \_\_\_\_\_ 条有向边; 设完全无向图中有  $n$  个顶点, 则该完全无向图中共有 \_\_\_\_\_ 条无向边。
3. 设关键字序列为  $(K_1, K_2, \dots, K_n)$ , 则用筛选法建初始堆必须从第 \_\_\_\_\_ 个元素开始进行筛选。
4. 解决散列表冲突的两种方法是 \_\_\_\_\_ 和 \_\_\_\_\_。
5. 设一棵三叉树中有 50 个度数为 0 的结点, 21 个度数为 2 的结点, 则该二叉树中度数为 3 的结点数有 \_\_\_\_\_ 个。
6. 高度为  $h$  的完全二叉树中最少有  $2^{h-1}$  \_\_\_\_\_ 个结点, 最多有  $2^h-1$  \_\_\_\_\_ 个结点。
7. 设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟直接插入排序结束后的结果是 \_\_\_\_\_ (12, 24, 35, 27, 18, 26) \_\_\_\_\_。

8. 设有一组初始关键字序列为 (24, 35, 12, 27, 18, 26), 则第 3 趟简单选择排序结束后的结果是\_\_\_\_ (12, 18, 24, 27, 35, 26)\_\_\_\_\_。
9. 设一棵二叉树的前序序列为 ABC, 则有\_\_\_\_\_5\_\_\_\_\_种不同的二叉树可以得到这种序列。
10. 下面程序段的功能是实现一趟快速排序, 请在下划线处填上正确的语句。

```

struct record {int key;datatype others;};
void quickpass(struct record r[], int s, int t, int &i)
{
    int j=t; struct record x=r[s]; i=s;
    while(i<j)
    {
        while (i<j && r[j].key>x.key) j=j-1;  if (i<j) {r[i]=r[j];i=i+1;}
        while (_____) i=i+1;  if (i<j) {r[j]=r[i];j=j-1;}
    }
    _____;
}

```

#### 四、算法设计题(20 分)

1. 设计在链式结构上实现简单选择排序算法。
2. 设计在顺序存储结构上实现求子串算法。
3. 设计求结点在二叉排序树中层次的算法。

### 数据结构试卷（七）

#### 一、选择题

- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. B | 2. B | 3. C | 4. B | 5. B  |
| 6. A | 7. C | 8. C | 9. B | 10. D |

#### 二、判断题

- |      |      |      |      |       |
|------|------|------|------|-------|
| 1. 对 | 2. 对 | 3. 对 | 4. 对 | 5. 对  |
| 6. 对 | 7. 对 | 8. 错 | 9. 错 | 10. 错 |

#### 三、填空题

1. s->left=p, p->right
2. n(n-1), n(n-1)/2
3. n/2
4. 开放定址法, 链地址法
5. 14
6.  $2^{h-1}$ ,  $2^h-1$
7. (12, 24, 35, 27, 18, 26)
8. (12, 18, 24, 27, 35, 26)
9. 5
10.  $i < j$  &&  $r[i].key < x.key$ ,  $r[i]=x$

#### 四、算法设计题

1. 设计在链式结构上实现简单选择排序算法。

```

void simpleselectsorlklst(lklist *&head)
{
    lklist *p,*q,*s;  int min,t;
    if(head==0 ||head->next==0) return;
    for(q=head; q!=0;q=q->next)
    {
        min=q->data; s=q;

```

- ```

        for(p=q->next; p!=0; p=p->next) if(min>p->data){ min=p->data; s=p;}
        if(s!=q){ t=s->data; s->data=q->data; q->data=t;}
    }
}

```
2. 设计在顺序存储结构上实现求子串算法。
- ```

void substring(char s[ ], long start, long count, char t[ ])
{
    long i,j,length=strlen(s);
    if (start<1 || start>length) printf("The copy position is wrong");
    else if (start+count-1>length) printf("Too characters to be copied");
    else { for(i=start-1,j=0; i<start+count-1;i++,j++) t[j]=s[i]; t[j]='\0';}
}

```
3. 设计求结点在二叉排序树中层次的算法。
- ```

int lev=0;
typedef struct node{int key; struct node *lchild,*rchild;}bitree;
void level(bitree *bt,int x)
{
    if (bt!=0)
        {lev++; if (bt->key==x) return; else if (bt->key>x) level(bt->lchild,x); else level(bt->rchild,x);}
}

```

## 数据结构试卷（八）

### 一、选择题(30 分)

- 字符串的长度是指（ ）。
  - 串中不同字符的个数
  - 串中不同字母的个数
  - 串中所含字符的个数
  - 串中不同数字的个数
- 建立一个长度为  $n$  的有序单链表的时间复杂度为（ ）。
  - $O(n)$
  - $O(1)$
  - $O(n^2)$
  - $O(\log_2 n)$
- 两个字符串相等的充要条件是（ ）。
  - 两个字符串的长度相等
  - 两个字符串中对应位置上的字符相等
  - 同时具备(A)和(B)两个条件
  - 以上答案都不对
- 设某散列表的长度为 100，散列函数  $H(k)=k \% P$ ，则  $P$  通常情况下最好选择（ ）。
  - 99
  - 97
  - 91
  - 93
- 在二叉排序树中插入一个关键字值的平均时间复杂度为（ ）。
  - $O(n)$
  - $O(\log_2 n)$
  - $O(n \log_2 n)$
  - $O(n^2)$
- 设一个顺序有序表  $A[1:14]$  中有 14 个元素，则采用二分法查找元素  $A[4]$  的过程中比较元素的顺序为（ ）。
  - $A[1], A[2], A[3], A[4]$
  - $A[1], A[14], A[7], A[4]$
  - $A[7], A[3], A[5], A[4]$
  - $A[7], A[5], A[3], A[4]$
- 设一棵完全二叉树中有 65 个结点，则该完全二叉树的深度为（ ）。
  - 8
  - 7
  - 6
  - 5
- 设一棵三叉树中有 2 个度数为 1 的结点，2 个度数为 2 的结点，2 个度数为 3 的结点，则该三叉链权中有（ ）个度数为 0 的结点。
  - 5
  - 6
  - 7
  - 8
- 设无向图  $G$  中的边的集合  $E=\{(a, b), (a, e), (a, c), (b, e), (e, d), (d, f), (f, c)\}$ ，则从顶点  $a$  出发进行深度优先遍历可以得到的一种顶点序列为（ ）。
  - aedfcb
  - acfebd
  - aebcfd
  - aedfbc

10. 队列是一种 ( ) 的线性表。  
(A) 先进先出 (B) 先进后出 (C) 只能插入 (D) 只能删除

## 二、判断题(20 分)

1. 如果两个关键字的值不等但哈希函数值相等, 则称这两个关键字为同义词。( )
2. 设初始记录关键字基本有序, 则快速排序算法的时间复杂度为  $O(n\log_2 n)$ 。( )
3. 分块查找的基本思想是首先在索引表中进行查找, 以便确定给定的关键字可能存在的块号, 然后在相应的块内进行顺序查找。( )
4. 二维数组和多维数组均不是特殊的线性结构。( )
5. 向二叉排序树中插入一个结点需要比较的次数可能大于该二叉树的高度。( )
6. 如果某个有向图的邻接表中第  $i$  条单链表为空, 则第  $i$  个顶点的出度为零。( )
7. 非空的双向循环链表中任何结点的前驱指针均不为空。( )
8. 不论线性表采用顺序存储结构还是链式存储结构, 删除值为  $X$  的结点的时间复杂度均为  $O(n)$ 。( )
9. 图的深度优先遍历算法中需要设置一个标志数组, 以便区分图中的每个顶点是否被访问过。( )
10. 稀疏矩阵的压缩存储可以用一个三元组表来表示稀疏矩阵中的非 0 元素。( )

## 三、填空题(30 分)

1. 设一组初始记录关键字序列为 (49, 38, 65, 97, 76, 13, 27, 50), 则以  $d=4$  为增量的一趟希尔排序结束后的结果为\_\_\_\_\_。
2. 下面程序段的功能是实现在二叉排序树中插入一个新结点, 请在下划线处填上正确的内容。  

```
typedef struct node{int data;struct node *lchild;struct node *rchild;}bitree;  
void bstinsert(bitree *&t,int k)  
{  
    if (t==0) {_____};t->data=k;t->lchild=t->rchild=0;  
    else if (t->data>k) bstinsert(t->lchild,k);else_____;  
}
```
3. 设指针变量  $p$  指向单链表中结点  $A$ , 指针变量  $s$  指向被插入的结点  $X$ , 则在结点  $A$  的后面插入结点  $X$  需要执行的语句序列:  $s->next=p->next$ ; \_\_\_\_\_;。
4. 设指针变量  $head$  指向双向链表中的头结点, 指针变量  $p$  指向双向链表中的第一个结点, 则指针变量  $p$  和指针变量  $head$  之间的关系是  $p=$ \_\_\_\_\_和  $head=$ \_\_\_\_\_ (设结点中的两个指针域分别为  $llink$  和  $rlink$ )。
5. 设某棵二叉树的中序遍历序列为 ABCD, 后序遍历序列为 BADC, 则其前序遍历序列为\_\_\_\_\_。
6. 完全二叉树中第 5 层上最少有\_\_\_\_\_个结点, 最多有\_\_\_\_\_个结点。
7. 设有向图中不存在有向边  $\langle V_i, V_j \rangle$ , 则其对应的邻接矩阵  $A$  中的数组元素  $A[i][j]$  的值等于\_\_\_\_\_。
8. 设一组初始记录关键字序列为 (49, 38, 65, 97, 76, 13, 27, 50), 则第 4 趟直接选择排序结束后的结果为\_\_\_\_\_。
9. 设连通图  $G$  中有  $n$  个顶点  $e$  条边, 则对应的最小生成树上有\_\_\_\_\_条边。
10. 设有一组初始记录关键字序列为 (50, 16, 23, 68, 94, 70, 73), 则将它们调整成初始堆只需把 16 与\_\_\_\_\_相互交换即可。

## 四、算法设计题(20 分)

1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。
2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

## 数据结构试卷(八) 参考答案

### 一、选择题

1. C    2. C    3. C    4. B    5. B  
6. C    7. B    8. C    9. A    10. A

### 二、判断题

1. 对    2. 错    3. 对    4. 错    5. 错
6. 对    7. 对    8. 对    9. 对    10. 对

### 三、填空题

1. (49, 13, 27, 50, 76, 38, 65, 97)
2. `t=(bitree *)malloc(sizeof(bitree)), bstinsert(t->rchild,k)`
3. `p->next=s`
4. `head->rlink, p->llink`
5. CABD
6. 1, 16
7. 0
8. (13, 27, 38, 50, 76, 49, 65, 97)
9.  $n-1$
10. 50

### 四、算法设计题

1. 设计一个在链式存储结构上统计二叉树中结点个数的算法。

```
void countnode(bitree *bt,int &count)
{
    if(bt!=0)
        {count++; countnode(bt->lchild,count); countnode(bt->rchild,count);}
}
```

2. 设计一个算法将无向图的邻接矩阵转为对应邻接表的算法。

```
typedef struct {int vertex[m]; int edge[m][m];}gadjmatrix;
typedef struct node1 {int info;int adjvertex; struct node1 *nextarc;}glinklistnode;
typedef struct node2 {int vertexinfo;glinklistnode *firstarc;}glinkheadnode;
void adjmatrixtoadjlist(gadjmatrix g1[ ],glinkheadnode g2[ ])
{
    int i,j; glinklistnode *p;
    for(i=0;i<=n-1;i++) g2[i].firstarc=0;
    for(i=0;i<=n-1;i++) for(j=0;j<=n-1;j++)
        if (g1.edge[i][j]==1)
        {
            p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=j;
            p->nextarc=g[i].firstarc; g[i].firstarc=p;
            p=(glinklistnode *)malloc(sizeof(glinklistnode));p->adjvertex=i;
            p->nextarc=g[j].firstarc; g[j].firstarc=p;
        }
}
```

## 数据结构试卷（九）

### 一、选择题(30 分)

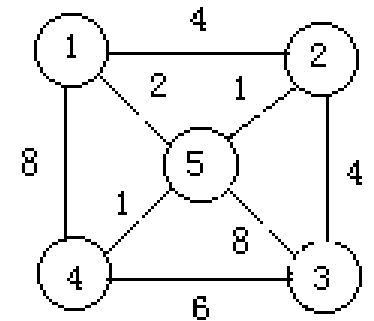
1. 下列程序段的时间复杂度为 ( )。  
`for(i=0; i<m; i++) for(j=0; j<t; j++) c[i][j]=0;`  
`for(i=0; i<m; i++) for(j=0; j<t; j++) for(k=0; k<n; k++) c[i][j]=c[i][j]+a[i][k]*b[k][j];`  
 (A)  $O(m*n*t)$       (B)  $O(m+n+t)$       (C)  $O(m+n*t)$       (D)  $O(m*t+n)$
2. 设顺序线性表中有  $n$  个数据元素，则删除表中第  $i$  个元素需要移动 ( ) 个元素。  
 (A)  $n-i$       (B)  $n+1-i$       (C)  $n-1-i$       (D)  $i$

3. 设 F 是由 T1、T2 和 T3 三棵树组成的森林，与 F 对应的二叉树为 B，T1、T2 和 T3 的结点数分别为 N1、N2 和 N3，则二叉树 B 的根结点的左子树的结点数为 ( )。  
(A)  $N1-1$  (B)  $N2-1$  (C)  $N2+N3$  (D)  $N1+N3$
4. 利用直接插入排序法的思想建立一个有序线性表的时间复杂度为 ( )。  
(A)  $O(n)$  (B)  $O(n\log_2 n)$  (C)  $O(n^2)$  (D)  $O(\log_2 n)$
5. 设指针变量 p 指向双向链表中结点 A，指针变量 s 指向被插入的结点 X，则在结点 A 的后面插入结点 X 的操作序列为 ( )。  
(A)  $p \rightarrow \text{right} = s; s \rightarrow \text{left} = p; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{right} = p \rightarrow \text{right};$   
(B)  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s;$   
(C)  $p \rightarrow \text{right} = s; p \rightarrow \text{right} \rightarrow \text{left} = s; s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right};$   
(D)  $s \rightarrow \text{left} = p; s \rightarrow \text{right} = p \rightarrow \text{right}; p \rightarrow \text{right} \rightarrow \text{left} = s; p \rightarrow \text{right} = s;$
6. 下列各种排序算法中平均时间复杂度为  $O(n^2)$  是 ( )。  
(A) 快速排序 (B) 堆排序 (C) 归并排序 (D) 冒泡排序
7. 设输入序列 1、2、3、...、n 经过栈作用后，输出序列中的第一个元素是 n，则输出序列中的第 i 个输出元素是 ( )。  
(A)  $n-i$  (B)  $n-1-i$  (C)  $n+1-i$  (D) 不能确定
8. 设散列表中有 m 个存储单元，散列函数  $H(\text{key}) = \text{key} \% p$ ，则 p 最好选择 ( )。  
(A) 小于等于 m 的最大奇数 (B) 小于等于 m 的最大素数  
(C) 小于等于 m 的最大偶数 (D) 小于等于 m 的最大合数
9. 设在一棵度数为 3 的树中，度数为 3 的结点数有 2 个，度数为 2 的结点数有 1 个，度数为 1 的结点数有 2 个，那么度数为 0 的结点数有 ( ) 个。  
(A) 4 (B) 5 (C) 6 (D) 7
10. 设完全无向图中有 n 个顶点，则该完全无向图中有 ( ) 条边。  
(A)  $n(n-1)/2$  (B)  $n(n-1)$  (C)  $n(n+1)/2$  (D)  $(n-1)/2$
11. 设顺序表的长度为 n，则顺序查找的平均比较次数为 ( )。  
(A) n (B)  $n/2$  (C)  $(n+1)/2$  (D)  $(n-1)/2$
12. 设有序表中的元素为 (13, 18, 24, 35, 47, 50, 62)，则在其中利用二分法查找值为 24 的元素需要经过 ( ) 次比较。  
(A) 1 (B) 2 (C) 3 (D) 4
13. 设顺序线性表的长度为 30，分成 5 块，每块 6 个元素，如果采用分块查找，则其平均查找长度为 ( )。  
(A) 6 (B) 11 (C) 5 (D) 6.5
14. 设有向无环图 G 中的有向边集合  $E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 4 \rangle \}$ ，则下列属于该有向图 G 的一种拓扑排序序列的是 ( )。  
(A) 1, 2, 3, 4 (B) 2, 3, 4, 1 (C) 1, 4, 2, 3 (D) 1, 2, 4, 3
15. 设有一组初始记录关键字序列为 (34, 76, 45, 18, 26, 54, 92)，则由这组记录关键字生成的二叉排序树的深度为 ( )。  
(A) 4 (B) 5 (C) 6 (D) 7

## 二、填空题(30 分)

1. 设指针 p 指向单链表中结点 A，指针 s 指向被插入的结点 X，则在结点 A 的前面插入结点 X 时的操作序列为：  
1)  $s \rightarrow \text{next} = \underline{\hspace{2cm}}$ ; 2)  $p \rightarrow \text{next} = s$ ; 3)  $t = p \rightarrow \text{data}$ ;  
4)  $p \rightarrow \text{data} = \underline{\hspace{2cm}}$ ; 5)  $s \rightarrow \text{data} = t$ ;
2. 设某棵完全二叉树中有 100 个结点，则该二叉树中有                      个叶子结点。
3. 设某顺序循环队列中有 m 个元素，且规定队头指针 F 指向队头元素的前一个位置，队尾指针 R 指向队尾元素的当前位置，则该循环队列中最多存储              队列元素。

4. 对一组初始关键字序列 (40, 50, 95, 20, 15, 70, 60, 45, 10) 进行冒泡排序, 则第一趟需要进行相邻记录的比较的次数为 \_\_\_\_\_, 在整个排序过程中最多需要进行 \_\_\_\_\_ 趟排序才可以完成。
5. 在堆排序和快速排序中, 如果从平均情况下排序的速度最快的角度来考虑应最好选择 \_\_\_\_\_ 排序, 如果从节省存储空间的角度来考虑则最好选择 \_\_\_\_\_ 排序。
6. 设一组初始记录关键字序列为 (20, 12, 42, 31, 18, 14, 28), 则根据这些记录关键字构造的二叉排序树的平均查找长度是 \_\_\_\_\_。
7. 设一棵二叉树的中序遍历序列为 BDCA, 后序遍历序列为 DBAC, 则这棵二叉树的前序序列为 \_\_\_\_\_。
8. 设用于通信的电文仅由 8 个字母组成, 字母在电文中出现的频率分别为 7、19、2、6、32、3、21、10, 根据这些频率作为权值构造哈夫曼树, 则这棵哈夫曼树的高度为 \_\_\_\_\_。
9. 设一组记录关键字序列为 (80, 70, 33, 65, 24, 56, 48), 则用筛选法建成的初始堆为 \_\_\_\_\_。
10. 设无向图 G (如右图所示), 则其最小生成树上所有边的权值之和为 \_\_\_\_\_。



### 三、判断题 (20 分)

1. 有向图的邻接表和逆邻接表中表结点的个数不一定相等。( )
2. 对链表进行插入和删除操作时不必移动链表中结点。( )
3. 子串“ABC”在主串“AABCABCD”中的位置为 2。( )
4. 若一个叶子结点是某二叉树的中序遍历序列的最后一个结点, 则它必是该二叉树的先序遍历序列中的最后一个结点。( )
5. 希尔排序算法的时间复杂度为  $O(n^2)$ 。( )
6. 用邻接矩阵作为图的存储结构时, 则其所占用的存储空间与图中顶点数无关而与图中边数有关。( )
7. 中序遍历一棵二叉排序树可以得到一个有序的序列。( )
8. 入栈操作和入队列操作在链式存储结构上实现时不需要考虑栈溢出的情况。( )
9. 顺序表查找指的是在顺序存储结构上进行查找。( )
10. 堆是完全二叉树, 完全二叉树不一定是堆。( )

### 五、算法设计题 (20 分)

1. 设计计算二叉树中所有结点值之和的算法。
2. 设计将所有奇数移到所有偶数之前的算法。
3. 设计判断单链表中元素是否是递增的算法。

## 数据结构试卷 (九) 参考答案

### 一、选择题

1. A    2. A    3. A    4. C    5. D
6. D    7. C    8. B    9. C    10. A
11. C    12. C    13. D    14. A    15. A

### 二、填空题

1.  $p \rightarrow next, s \rightarrow data$
2. 50
3.  $m-1$
4. 6, 8
5. 快速, 堆
6. 19/7
7. CBDA
8. 6

9. (24, 65, 33, 80, 70, 56, 48)

10. 8

### 三、判断题

1. 错    2. 对    3. 对    4. 对    5. 错

6. 错    7. 对    8. 对    9. 错    10. 对

#### 四、算法设计题

1. 设计计算二叉树中所有结点值之和的算法。

```
void sum(bitree *bt,int &s)
{
    if(bt!=0) {s=s+bt->data; sum(bt->lchild,s); sum(bt->rchild,s);}
}
```

2. 设计将所有奇数移到所有偶数之前的算法。

```
void quickpass(int r[], int s, int t)
{
    int i=s,j=t,x=r[s];
    while(i<j)
    {
        while (i<j && r[j]%2==0) j=j-1;    if (i<j) {r[i]=r[j];i=i+1;}
        while (i<j && r[i]%2==1) i=i+1;    if (i<j) {r[j]=r[i];j=j-1;}
    }
    r[i]=x;
}
```

3. 设计判断单链表中元素是否是递增的算法。

```
int isriselk(lklist *head)
{
    if(head==0||head->next==0) return(1);else
    for(q=head,p=head->next; p!=0; q=p,p=p->next)if(q->data>p->data) return(0);
    return(1);
}
```

## 数据结构试卷（十）

### 一、选择题(24 分)

1. 下列程序段的时间复杂度为 ( )。

```
i=0, s=0; while (s<n) {s=s+i; i++; }
```

(A)  $O(n_{1/2})$                       (B)  $O(n_{1/3})$                       (C)  $O(n)$                       (D)  $O(n_2)$

2. 设某链表中最常用的操作是在链表的尾部插入或删除元素, 则选用下列 ( ) 存储方式最节省运算时间。

(A) 单向链表 (B) 单向循环链表

(C) 双向链表 (D) 双向循环链表

3. 设指针 q 指向单链表中结点 A，指针 p 指向单链表中结点 A 的后继结点 B，指针 s 指向被插入的结点 X，则在结点 A 和结点 B 插入结点 X 的操作序列为（ ）。

(A)  $s \rightarrow next = p \rightarrow next$ ;  $p \rightarrow next = s$ ; (B)  $q \rightarrow next = s$ ;  $s \rightarrow next = p$ ;

(C) p->next=s->next; s->next=p;      (D) p->next=s; s->next=q;

4. 设输入序列为 1、2、3、4、5、6，则通过栈的作用后可以得到的输出序列为（ ）。

(A) 5, 3, 4, 6, 1, 2                      (B) 3, 2, 5, 6, 4, 1

(C) 3, 1, 2, 5, 4, 6                      (D) 1, 5, 4, 6, 2, 3



5. 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连续的 55 个存储单元中, 每个数组元素占 1 个字节的存储空间, 则 A[5][4]地址与 A[0][0]的地址之差为 ( )。
- (A) 10 (B) 19 (C) 28 (D) 55
6. 设一棵 m 叉树中有  $N_1$  个度数为 1 的结点,  $N_2$  个度数为 2 的结点, …….,  $N_m$  个度数为 m 的结点, 则该树中共有 ( ) 个叶子结点。
- (A)  $\sum_{i=1}^m (i-1)N_i$  (B)  $\sum_{i=1}^m N_i$  (C)  $\sum_{i=2}^m N_i$  (D)  $1 + \sum_{i=2}^m (i-1)N_i$
7. 二叉排序树中左子树上所有结点的值均 ( ) 根结点的值。
- (A) < (B) > (C) = (D) !=
8. 设一组权值集合 W=(15, 3, 14, 2, 6, 9, 16, 17), 要求根据这些权值集合构造一棵哈夫曼树, 则这棵哈夫曼树的带权路径长度为 ( )。
- (A) 129 (B) 219 (C) 189 (D) 229
9. 设有 n 个关键字具有相同的 Hash 函数值, 则用线性探测法把这 n 个关键字映射到 HASH 表中需要做 ( ) 次线性探测。
- (A)  $n^2$  (B)  $n(n+1)$  (C)  $n(n+1)/2$  (D)  $n(n-1)/2$
10. 设某棵二叉树中只有度数为 0 和度数为 2 的结点且度数为 0 的结点数为 n, 则这棵二叉中共有 ( ) 个结点。
- (A) 2n (B) n+1 (C) 2n-1 (D) 2n+1
11. 设一组初始记录关键字的长度为 8, 则最多经过 ( ) 趟插入排序可以得到有序序列。
- (A) 6 (B) 7 (C) 8 (D) 9
12. 设一组初始记录关键字序列为(Q, H, C, Y, P, A, M, S, R, D, F, X), 则按字母升序的第一趟冒泡排序结束后的结果是 ( )。
- (A) F, H, C, D, P, A, M, Q, R, S, Y, X  
 (B) P, A, C, S, Q, D, F, X, R, H, M, Y  
 (C) A, D, C, R, F, Q, M, S, Y, P, H, X  
 (D) H, C, Q, P, A, M, S, R, D, F, X, Y

## 二、填空题(48 分, 其中最后两小题各 6 分)

- 设需要对 5 个不同的记录关键字进行排序, 则至少需要比较 \_\_\_\_\_ 次, 至多需要比较 \_\_\_\_\_ 次。
- 快速排序算法的平均时间复杂度为 \_\_\_\_\_, 直接插入排序算法的平均时间复杂度为 \_\_\_\_\_。
- 设二叉排序树的高度为 h, 则在该树中查找关键字 key 最多需要比较 \_\_\_\_\_ 次。
- 设在长度为 20 的有序表中进行二分查找, 则比较一次查找成功的结点数有 \_\_\_\_\_ 个, 比较两次查找成功有结点数有 \_\_\_\_\_ 个。
- 设一棵 m 叉树的结点数 n, 用多重链表表示其存储结构, 则该树中有 \_\_\_\_\_ 个空指针域。
- 设指针变量 p 指向单链表中结点 A, 则删除结点 A 的语句序列为:  
 $q=p \rightarrow next$ ;  $p \rightarrow data=q \rightarrow data$ ;  $p \rightarrow next=$  \_\_\_\_\_;  $free(q)$ ;
- 数据结构从逻辑上划分为三种基本类型: \_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
- 设无向图 G 中有 n 个顶点 e 条边, 则用邻接矩阵作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 \_\_\_\_\_; 用邻接表作为图的存储结构进行深度优先或广度优先遍历的时间复杂度为 \_\_\_\_\_。
- 设散列表的长度为 8, 散列函数  $H(k)=k \% 7$ , 用线性探测法解决冲突, 则根据一组初始关键字序列 (8, 15, 16, 22, 30, 32) 构造出的散列表的平均查找长度是 \_\_\_\_\_。
- 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 则第 3 趟冒泡排序结束后的结果为 \_\_\_\_\_。

11. 设一组初始关键字序列为 (38, 65, 97, 76, 13, 27, 10), 则第 3 趟简单选择排序后的结果为\_\_\_\_\_。
12. 设有向图 G 中的有向边的集合  $E=\{<1, 2>, <2, 3>, <1, 4>, <4, 5>, <5, 3>, <4, 6>, <6, 5>\}$ , 则该图的一个拓扑序列为\_\_\_\_\_。
13. 下面程序段的功能是建立二叉树的算法, 请在下划线处填上正确的内容。
- ```
typedef struct node{int data;struct node  *lchild;_____;}bitree;
void createbitree(bitree *&bt)
{
    scanf("%c",&ch);
    if(ch=='#') _____;else
        { bt=(bitree*)malloc(sizeof(bitree)); bt->data=ch; _____;createbitree(bt->rchild);}
}
```
14. 下面程序段的功能是利用从尾部插入的方法建立单链表的算法, 请在下划线处填上正确的内容。
- ```
typedef struct node {int data; struct node *next;} lklist;
void lklistcreate(_____ *&head )
{
    for (i=1;i<=n;i++)
    {
        p=(lklist *)malloc(sizeof(lklist));scanf("%d",&(p->data));p->next=0;
        if(i==1)head=q=p;else {q->next=p;_____;}
    }
}
```

### 三、算法设计题 (22 分)

- 设计在链式存储结构上合并排序的算法。
- 设计在二叉排序树上查找结点 X 的算法。
- 设关键字序列  $(k_1, k_2, \dots, k_{n-1})$  是堆, 设计算法将关键字序列  $(k_1, k_2, \dots, k_{n-1}, x)$  调整为堆。

## 数据结构试卷（十）参考答案

### 一、选择题

1. A    2. D    3. B    4. B    5. B    6. D  
7. A    8. D    9. D    10. C    11. B    12. D

### 二、填空题

- 4, 10
- $O(n\log_2 n)$ ,  $O(n^2)$
- n
- 1, 2
- $n(m-1)+1$
- q->next
- 线性结构, 树型结构, 图型结构
- $O(n^2)$ ,  $O(n+e)$
- 8/3
- (38, 13, 27, 10, 65, 76, 97)
- (10, 13, 27, 76, 65, 97, 38)
- 124653
- struct node \*rchild, bt=0, createbitree(bt->lchild)
- lklist, q=p

### 三、算法设计题

1. 设计在链式存储结构上合并排序的算法。

```
void mergelklist(lklist *ha,lklist *hb,lklist *&hc)
{
    lklist *s=hc=0;
    while(ha!=0 && hb!=0)
        if(ha->data<hb->data){ if(s==0) hc=s=ha; else { s->next=ha; s=ha; };ha=ha->next;}
        else { if(s==0) hc=s=hb; else { s->next=hb; s=hb; };hb=hb->next;}
    if(ha==0) s->next=hb; else s->next=ha;
}
```

2. 设计在二叉排序树上查找结点 X 的算法。

```
bitree *bstsearch1(bitree *t, int key)
{
    bitree *p=t;
    while(p!=0) if (p->key==key) return(p);else if (p->key>key)p=p->lchild; else p=p->rchild;
    return(0);
}
```

3. 设关键字序列  $(k_1, k_2, \dots, k_{n-1})$  是堆, 设计算法将关键字序列  $(k_1, k_2, \dots, k_{n-1}, x)$  调整为堆。

```
void adjustheap(int r[],int n)
{
    int j=n,i=j/2,temp=r[j-1];
    while (i>=1) if (temp>=r[i-1])break; else{r[j-1]=r[i-1]; j=i; i=i/2;}
    r[j-1]=temp;
}
```