

## 一、选择题 (20 分)

1. 组成数据的基本单位是 ( )。  
(A) 数据项 (B) 数据类型 (C) 数据元素 (D) 数据变量
2. 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 ( )。  
(A) 线性结构 (B) 树型结构 (C) 图型结构 (D) 集合
3. 数组的逻辑结构不同于下列 ( ) 的逻辑结构。  
(A) 线性表 (B) 栈 (C) 队列 (D) 树
4. 二叉树中第  $i(i \geq 1)$  层上的结点数最多有 ( ) 个。  
(A)  $2i$  (B)  $2^i$  (C)  $2^{i-1}$  (D)  $2i-1$
5. 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)  $p \rightarrow next = p \rightarrow next \rightarrow next$  (B)  $p = p \rightarrow next$   
(C)  $p = p \rightarrow next \rightarrow next$  (D)  $p \rightarrow next = p$
6. 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 ( )。  
(A) 6 (B) 4 (C) 3 (D) 2
7. 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 ( )。  
(A) 100 (B) 40 (C) 55 (D) 80
8. 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  $B$  的度数为 ( )。  
(A) 3 (B) 4 (C) 5 (D) 1
9. 根据二叉树的定义可知二叉树共有 ( ) 种不同的形态。  
(A) 4 (B) 5 (C) 6 (D) 7
10. 设有以下四种排序方法, 则 ( ) 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)

1. 1. 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  $F = \underline{\hspace{2cm}}$ ;
2. 2. 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  $\underline{\hspace{2cm}}$ 。
3. 3. 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。

4. 4. 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\_\_\_\_\_。
5. 5. 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\_\_\_\_\_个表头结点和\_\_\_\_\_个表结点。
6. 6. 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\_\_\_\_\_关系。
7. 7. 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\_\_\_\_\_。
8. 8. 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\_\_\_\_\_, 编号为 8 的左孩子结点的编号是\_\_\_\_\_。
9. 9. 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。  

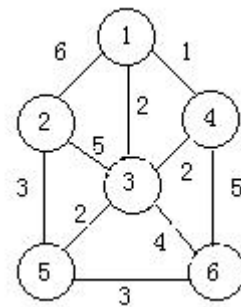
```

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. 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。

### 三、应用题 (30 分)

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。



#### 四、算法设计题(20 分)

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

## 数据结构试卷参考答案

### 一、选择题

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

### 二、填空题

1. 1.  $(F+1) \% m$   
2. 2.  $O(n), O(n)$   
3. 3.  $2n, n+1$   
4. 4.  $s->next=p->next; s->next=s$   
5. 5.  $n, 2e$   
6. 6.  $m=2e$   
7. 7. CBA  
8. 8. 4, 16  
9. 9.  $i-j+1, 0$   
10. 10.  $n-1$

### 三、应用题

1. 1. 链式存储结构略，前序 ABDEC，中序 DBEAC，后序 DEBCA。  
2. 2. 哈夫曼树略，WPL=78  
3. 3. (18,5,16,19,21,23), (5, 16, 21, 19, 18, 23)  
4. 4. 线性探测： $\wedge$  8  $\wedge$  10 25 32 27 68    链地址法： $h_6 - > 27$   
5. 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. 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. 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. 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);}
}

```