

125  
1600元

# 《数据结构》

院(系、专业) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

请考生注意:

1. 答案请写在答题纸上, 写在试卷上一律无效!
2. 考试完毕, 请将答题纸和试卷交给监考老师, 不得带出考场!

装  
订  
线  
内  
不  
要  
答  
题

得分

## 一、填空题 (每小题 2 分, 共 20 分)

1. 数据的四种基本逻辑结构有树形结构、线性结构、\_\_\_\_\_结构和集合结构。
2. 散列表中,  $H$  为散列函数, 若  $H(42) = H(12)$ , 则 42, 12 相对于散列函数  $H$  而言称为\_\_\_\_\_。
3. 堆栈中 Top 指向栈顶元素, 判断堆栈为空的条件是\_\_\_\_\_。
4. 利用 AOE 网进行工程安排, 完成工程所需的最短时间是指从开始结点到完成结点的\_\_\_\_\_路径的长度, 这条路径被称为关键路径。
5. 若对某算法求得关键步骤执行次数  $f(n) = 20 + 30\log_2 n + 40n$ , 则其渐近时间复杂度为  $O(\underline{\quad})$ 。
6. 设森林  $F$  对应的二叉树为  $B$ , 它有  $n$  个结点,  $B$  的根为  $r$ ,  $r$  的右子树结点个数为  $m$ , 则  $F$  中第一棵树的结点个数是\_\_\_\_\_。
7.  $5 \times 8$  的二维数组按行优先存储, 第一个数组元素  $a[0][0]$  的存储地址是 100, 每个元素占有 2 个存储单元, 则数组元素  $a[3][6]$  的存储地址是\_\_\_\_\_。
8. 线性表采用对半搜索必须满足两个条件: (1) 线性表必须是\_\_\_\_\_; (2) 存储结构必须采用顺序存储结构。
9. 在含  $n$  个顶点和  $e$  条边的有向图的邻接矩阵中, 零元素的个数为\_\_\_\_\_。
10. 若某二叉树的后序遍历序列为: CDAEF, 此二叉树的根结点是\_\_\_\_\_。

得分

## 二、选择题 (每小题 2 分, 共 20 分)

1. 微博用户 A 关注了 B, B 关注了 A, C 也关注 B, 则表示这种关注关系最合适的数据结构应该是: \_\_\_\_\_。  
A. 树      B. 图      C. 散列表      D. 二叉树

2、以下关键字序列\_\_\_\_\_是一个最小堆。

- A. 12, 72, 31, 25, 99, 58                      B. 99, 25, 31, 72, 12, 58  
C. 12, 58, 25, 99, 31, 72                      D. 12, 25, 58, 31, 99, 72

3、后缀表达式:  $8 \ 4 \ 4 \ + \ / \ 3 \ 3 \ * \ +$  的值为\_\_\_\_\_。

- A. 10                      B. 12  
C. 7                      D. 以上答案都不正确

4、对稀疏矩阵采用三元组方式存储的最主要目的是\_\_\_\_\_。

- A. 使表达变得简单                      B. 使矩阵元素的存取变得简单  
C. 去掉矩阵中的多余元素              D. 压缩存储空间

5、冒泡排序在最好的情况下的渐近时间复杂度是\_\_\_\_\_。

- A.  $O(n)$                       B.  $O(\log n)$                       C.  $O(n^2)$                       D.  $O(n \log n)$

6、已知一棵完全二叉树的第6层(设根为第1层)有8个叶结点,则该完全二叉树的结点个最多是\_\_\_\_\_。

- A. 52                      B. 111                      C. 119                      D. 127

7、对半搜索有序表(10,30,36,41,52,54,66,73,84,97),在表中搜索关键字34,则它将依次与表中\_\_\_\_\_比较大小,最终搜索失败。

- A. 52,41,30                      B. 54,84,66  
C. 52,30,36                      D. 54,73,66

8、二叉搜索树中,关键字最大的结点\_\_\_\_\_。

- A. 左子树一定为空                      B. 右子树一定为空  
C. 左右子树均为空                      D. 左右子树均不为空

9、如果线性表最常用的操作是读取第*i*个元素的值,则采用\_\_\_\_\_存储方式最节省时间。

- A. 顺序表                      B. 带头结点的单链表  
C. 不带表头结点的单链表              D. 双向链表

10、设有向图G的边集为 $\langle 0,1 \rangle, \langle 1,2 \rangle, \langle 4,1 \rangle, \langle 4,5 \rangle, \langle 5,3 \rangle, \langle 2,3 \rangle$ ,则下面不属于该图的拓扑排序的是\_\_\_\_\_。

- A. 041253                      B. 402153                      C. 045123                      D. 401523

得分

### 三、简答题(每小题8分,共48分)

1、如图1所示顶点表示小区,边表示连接小区间的光纤,边上的权表示铺设光纤需花费的造价。现在需要让小区间光纤通信畅通且总造价最省,请画出方案构建过程,并给出相应的总造价。

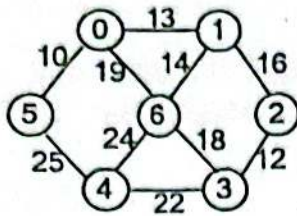


图1

2、对长度为 13 的有序表（假定下标从 0 开始标记）进行对半搜索，请画出对应的二叉判定树。

3、请将图 2 所示的森林转成二叉树。

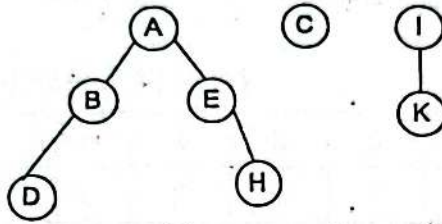


图2

4、已知散列表如下所示，长度  $M=11$ ，依次输入关键字 17, 18, 37, 74，试以双散列法解决冲突，散列函数为  $h_1(\text{key})=\text{key}\%11$ ， $h_2(\text{key})=\text{key}\%9+1$ ，请填写散列表。

下标	0	1	2	3	4	5	6	7	8	9	10
元素						5	6		41		

5、(1) 从图 3 所示的 3 阶 B-树中插入 35，画出插入新元素后的 B-树。

(2) 在 (1) 所得 3 阶 B-树的基础上再插入 47，画出插入新元素后的 B-树。

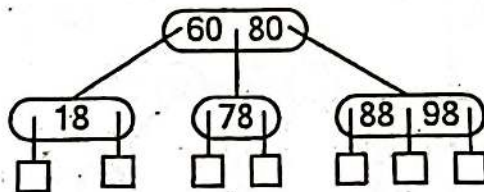


图3

6、已知一组序列为：55, 80, 66, 40, 42, 88。

(1) 请给出第一趟快速排序的结果。

(2) 请写出最坏情况下，快速排序的时间复杂度。

得分

#### 四、算法设计题（每小题 6 分，共 12 分）

1、二叉搜索树 T 用二叉链表存储结构表示，编写算法按递减顺序打印 T 中元素关键字的值。

相关结构体定义如下：

```
typedef struct elemtype {
    int Key;
    char Data;
} ElemType;
typedef struct bstnode {
    ElemType Element;
    struct bstnode *lchild;
    struct bstnode *rchild;
}BSTNode;
```

2、不带表头结点的单链表中元素各不相等，编写算法找出所有元素的最大值与链表第一个结点的值进行交换，如果成功，返回1；否则，返回0。

相关结构体定义如下：

```
typedef struct node
{
    ElemType element;
    struct node *link;
}Node;
```

```
typedef struct singleList
{
    struct node * first;
    int n;
}SingleList;
```

# 《数据结构》

## 一、填空题 (每小题2分, 共20分)

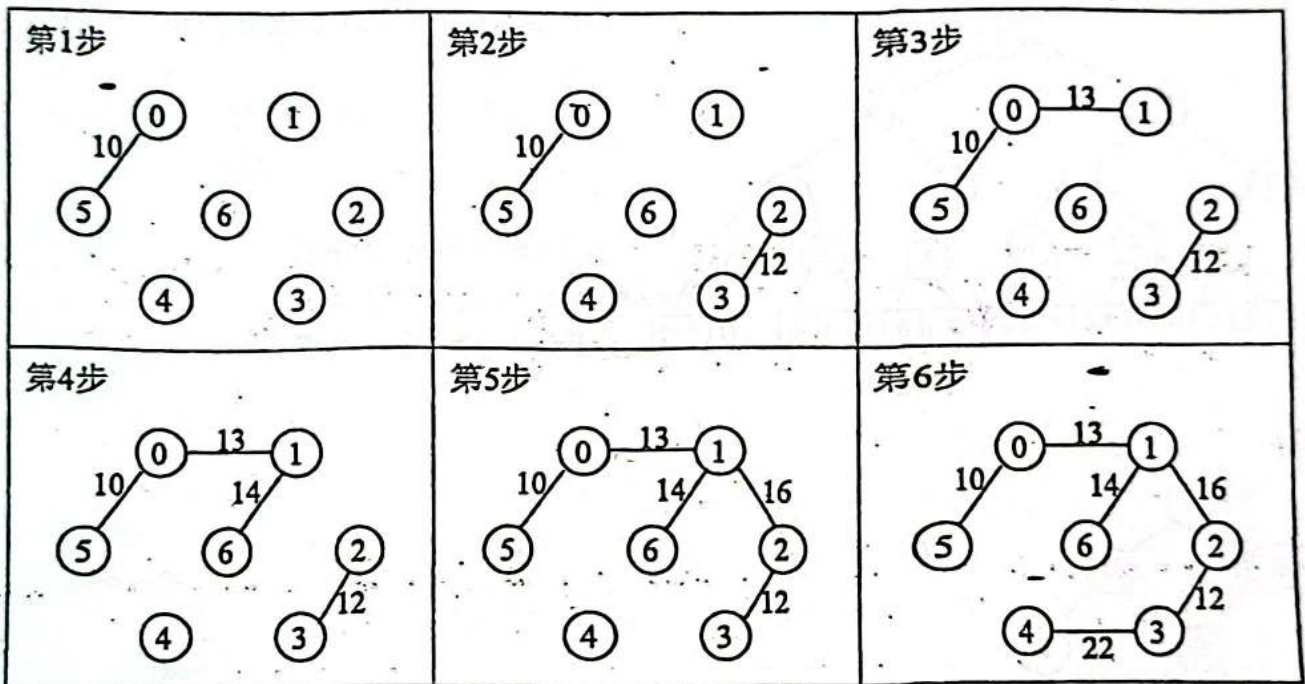
题号	1	2	3	4	5	6	7	8	9	10
答案	图形	同义词	Top=-1	最长	$n \log_2 n$	$n-m$	160	有序	$n^2-e$	F

## 二、单项选择题 (每小题2分, 共20分)

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

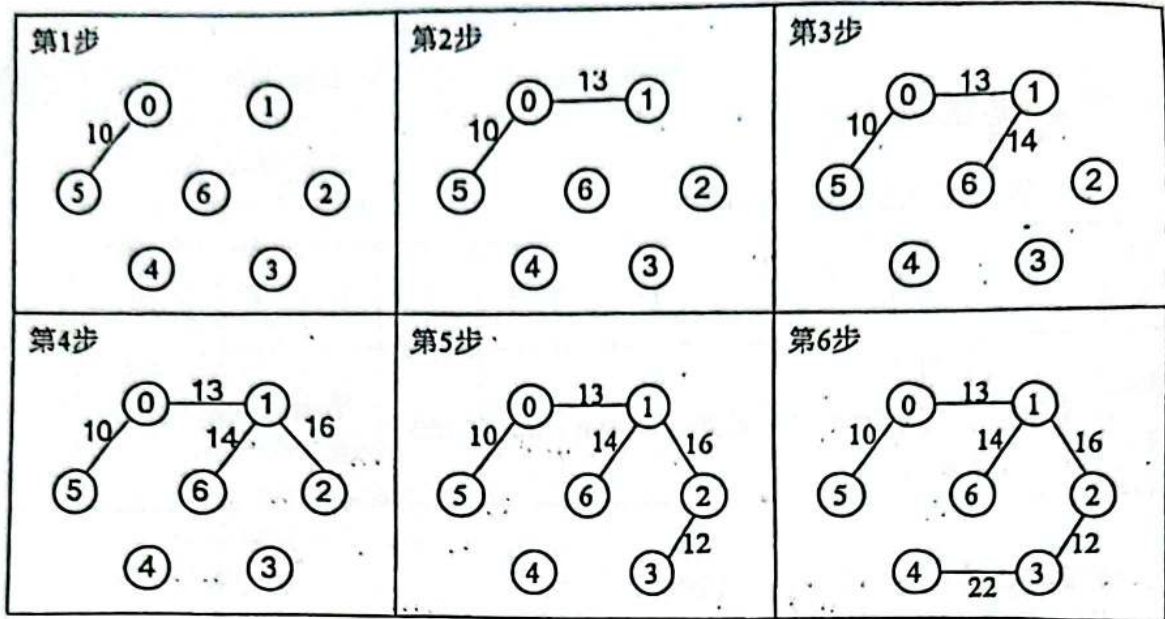
## 三、简答题 (每小题8分, 共40分)

1、



或者

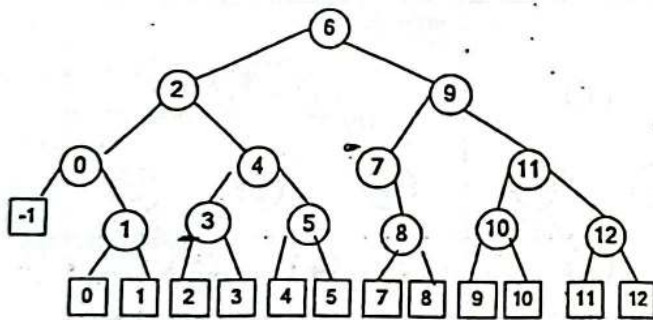
参考答案 第 1 页 共 4 页



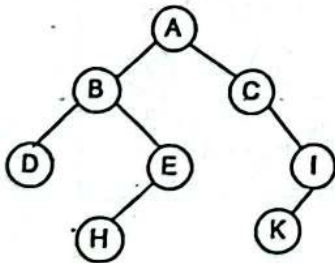
最小代价生成树的代价为87

每步1分，代价计算正确得2分

2、8分



3、8分

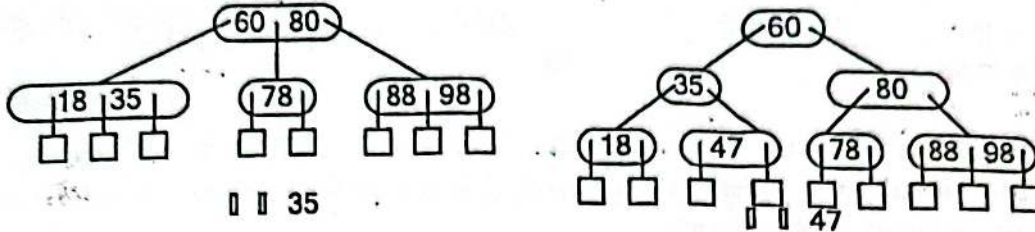


参考答案 第2页共4页

4、每个2分

下标	0	1	2	3	4	5	6	7	8	9	10
元素	74				17	5	6	18	41		37

5、每个4分



6、

第一趟排序后序列为: 40, 42, 55, 66, 80, 88(6分)

$O(n^2)$  (2分)

#### 四、算法设计题 (共12分)

1、6分

```

Void ReduceOrder (BSTNode * p)
{
    if (!p) return;          2分
    else
    {
        reduceOrder(p->rchild);  4分
        printf("%d", p->Element.Key);
        ReduceOrder (p->lchild);
    }
}
    
```

2、6分

```

int Swap()
{
    if (!first)
        return 0;
    Node *p, *max;
    ElemType temp;
    
```

```
p= first-> link;           2分
max = first;
while(p)                   2分
{
    if(p->element >max->element)
        max = p;
    p = p->link;
}
if(max != p)               2分
{
    temp = first->element;
    first->element = max->element;
    max->element = temp;
}
return 1;
}
```



# 《数据结构》

院(系、专业) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

请考生注意:

1. 答案请写在答题纸上, 写在试卷上一律无效!
2. 考试完毕, 请将答题纸和试卷交给监考老师, 不得带出考场!

得分

## 一、填空题 (每小题 2 分, 共 20 分)

1. 线性表的链式存储结构中, 通过 \_\_\_\_\_ 表示元素之间的关系。
2. 对  $n$  个元素的序列执行冒泡排序最好情况下算法需要执行 \_\_\_\_\_ 趟排序。
3. AOE 网中的关键路径是源点到汇点之间 \_\_\_\_\_ 路径。
4. 以 38, 45, 93, 28, 14, 79, 59 为输入序列建立二叉搜索树, 在此二叉搜索树中删除 79 后, 此二叉搜索树的根结点为 \_\_\_\_\_。
5. 下列程序段中划线语句的渐近时间复杂度为 \_\_\_\_\_。  

```
i=1; y=0;  
do{  
  y++; i=2*i;  
}while(i<n).
```
6.  $a*(b+c)-d$  的后缀表达式是 \_\_\_\_\_。
7. 不带表头结点的单链表中, first 为头指针, 当 \_\_\_\_\_ 时, 不带表头结点的单链表为空。
8. 已知高度为 3 的二叉平衡树中至少有 4 个结点, 高度为 4 的二叉平衡树上至少有 7 个结点, 则高度为 5 的二叉平衡树上至少有 \_\_\_\_\_ 结点。
9. 在含  $n$  个顶点和  $e$  条边的无向图的邻接矩阵中, 零元素的个数为 \_\_\_\_\_。
10. 一棵有  $n$  个结点的满二叉树有 \_\_\_\_\_ 个叶子结点。

得分

二、选择题（每小题 2 分，共 20 分）

- 1、将森林 F 转换为对应的二叉树 T, F 中叶结点的个数等于\_\_\_\_\_。
- A. T 中叶结点的个数                      B. T 中度为 1 的结点个数
- C. T 中左孩子指针为空的结点个数        D. T 中左孩子指针为空的结点个数
- 2、设顺序表长度为 n, 则在位置  $i(i = -1, 0, 1, 2, \dots, n-1)$  后插入元素需要移动\_\_\_\_\_个元素。
- A.  $n-i-1$                       B.  $n-i$                       C.  $n-i+1$                       D.  $n-i+2$
- 3、以下关键字序列\_\_\_\_\_是一个最小堆。
- A. 10, 72, 31, 25, 99, 58                      B. 99, 25, 31, 72, 10, 58
- C. 10, 58, 25, 99, 31, 72                      D. 10, 25, 58, 31, 99, 72
- 4、已知图的边集合  $E(G) = \{ \langle 6,1 \rangle, \langle 1,2 \rangle, \langle 4,1 \rangle, \langle 4,5 \rangle, \langle 5,3 \rangle, \langle 2,3 \rangle \}$ , 则序列\_\_\_\_\_是该图的拓扑序列之一。
- A. 6, 3, 4, 5, 1, 2                      B. 6, 1, 2, 3, 4, 5                      C. 4, 5, 6, 1, 2, 3                      D. 4, 3, 5, 2, 1, 6
- 5、设循环队列的元素存放在一维数组  $Q[30]$  中, 队列非空时, front 指示队列首结点的前一个位置, rear 指示队列尾结点。如果队列中元素的个数为 10, front 的值为 25, 则 rear 应指向的元素是\_\_\_\_\_。
- A.  $Q[4]$                       B.  $Q[5]$                       C.  $Q[14]$                       D.  $Q[15]$
- 6、对半搜索有序表 (10,30,36,41,52,54,66,73,84,97), 在表中搜索关键字 34, 则它将进行\_\_\_\_\_次比较最终搜索失败。
- A. 2                      B. 3                      C. 4                      D. 5
- 7、以 55, 30, 17, 12, 16, 13 建立二叉平衡树, 此二叉平衡树的根结点为\_\_\_\_\_。
- A. 13                      B. 16                      C. 30                      D. 55
- 8、若从无向图的任意一个顶点出发进行深度优先遍历可以访问图中所有的顶点, 则该图一定是\_\_\_\_\_。
- A. 非连通图                      B. 连通图                      C. 强连通图                      D. 完全图
- 9、利用哈夫曼算法构造出来的树是\_\_\_\_\_。
- A. 二叉排序树                      B. 扩充二叉树                      C. AVL 树                      D. 完全二叉树
- 10、若数据元素序列 10, 12, 13, 7, 8, 9, 33, 2, 5 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\_\_\_\_\_。
- A. 冒泡排序                      B. 快速排序                      C. 简单选择排序                      D. 直接插入排序

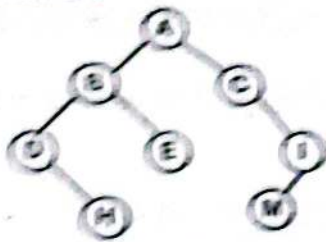
得分

三、简答题（每小题 8 分，共 48 分）

- 1、设一个散列表的长度  $M = 11$ , 散列函数是  $H(key) = key \% 11$ , 现采用二次探查法解决冲突。已知散列表如下所示, 依次插入关键字 3、25、54, 34, 请填写散列表。

下标	0	1	2	3	4	5	6	7	8	9	10
元素				80	48						43

2. 将题2图所示的二叉树转换为森林。



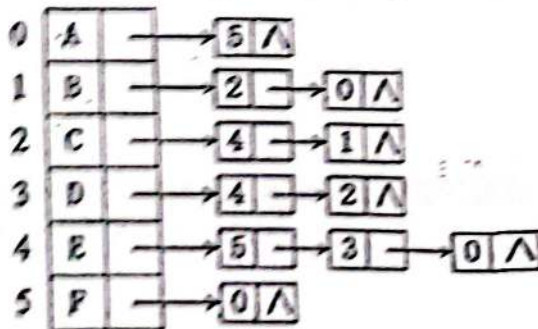
题2图

3. 从题3图所示的4阶B-树中依次插入41, 51, 分别画出插入新元素后的B-树。



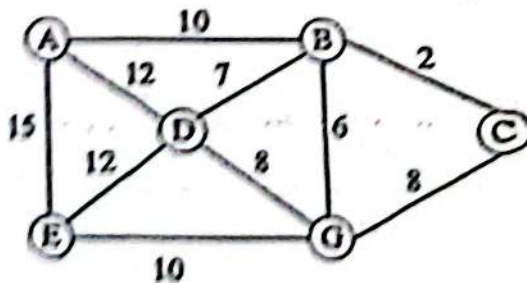
题3图

4. 已知图G的邻接表如题4图所示, 若以顶点B为出发点, 请分别写出深度优先搜索和广度优先搜索的顶点序列。



题4图

5. 如题5图所示为一个地区的交通网, 顶点表示城市, 边表示连接城市间的公路, 边上的权表示修建公路需花费的造价。现在需要选择能够沟通每个城市且总造价最省的5条公路, 请画出所有可能的方案, 并给出该工程的总造价。



题5图

6、对元素序列49, 38, 66, 82, 13, 53, 3按教材中冒泡排序算法进行排序。

(1) 写出第一趟冒泡排序的结果。

(2) 写出最好、最坏和平均情况下冒泡排序的渐近时间复杂度。

得分

#### 四、算法设计题（每小题6分，共12分）

1、设二叉搜索树以二叉链表结构表示，结点结构体定义为：

```
typedef struct {
    int element;
    BSTNode *lchild;
    BSTNode *rchild;
}BSTNode;
```

试设计算法输出二叉搜索树中的最大值。

```
void PrintMax (BSTNode *r) //r 为二叉搜索树的根结点
{
    ...
}
```

2、已知单链表中的元素有序非递减排列，请编写算法删除表中所有值相同的元素。本结构体定义如下：

```
typedef struct node
{
    ElemType element;
    struct node *link;
}Node;
```

```
typedef struct singleList
{
    struct node * first;
    int n;
}SingleList;
```

```
void delete(SingleList *L)
{
    ...
}
```

## 《数据结构》

### 一、填空题 (每小题2分, 共20分)

题号	1	2	3	4	5	6	7	8	9	10
答案	结点指针或指针	1	最长	38	$O(\log_2 n)$	$abc+d$	first = = NULL	12	$n^2-2e$	$(n+1)/2$

### 二、单项选择题 (每小题2分, 共20分)

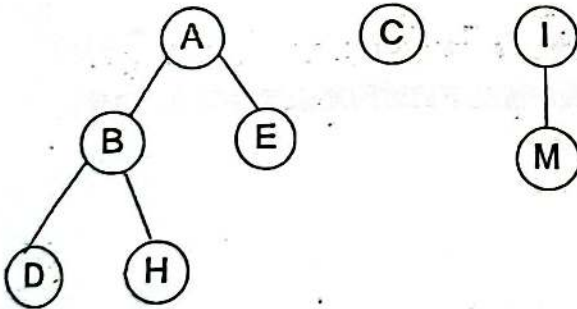
题号	1	2	3	4	5	6	7	8	9	10
答案	C	A	D	C	B	B	B	B	B	D

### 三、简答题 (每小题8分, 共40分)

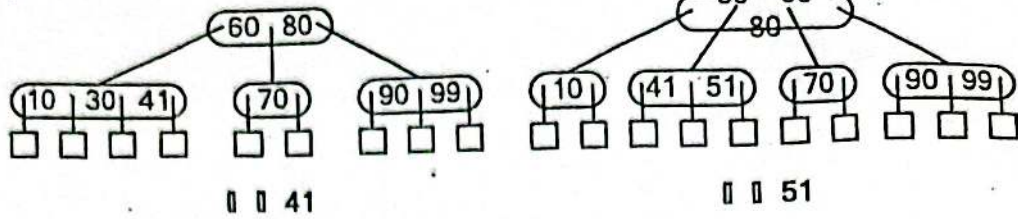
1、

下标	0	1	2	3	4	5	6	7	8	9	10
元素	54	34	3	80	48			25			-43

2、8分



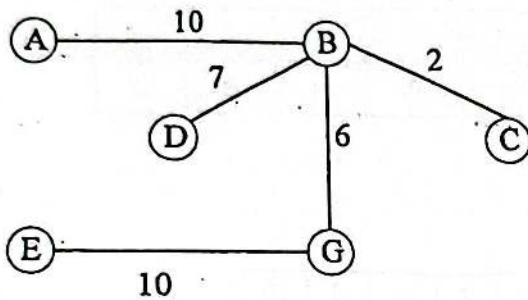
3、各4分



4、

- (1) 深度优先搜索序列: BCEFAD (4分)  
 (2) 宽度优先搜索序列: BCAEFD (4分)

5、(1) 该工程只有一种可能的方案



(6分)

(2) 该工程的总造价为:  $10+2+7+6+10=35$ . (2分)

6、

- (1) 第一趟冒泡排序的结果: 38 49 66 13 53 3 82 (5分)  
 (2) 平均和最坏情况的时间复杂度为 $O(n^2)$ ; 最好情况下的时间复杂度为 $O(n)$ . (3分)

#### 四、算法设计题 (共12分)

1、6分

```
void PrintMax (BSTNode *r)
```

```
{
```

```
    int max;
```

```
    BSTNode *p;
```

```
    p=r; (2分)
```

```
    while(p->rChild) (2分)
```

```
        p=p->rChild;
```

参考答案 第 2 页 共 3 页

```
max = p->element; //获得最大值 (2分)
printf("最大值是 %d\n", max);
```

```
)
```

2、6分

```
void delete(SingleList *L)
```

```
{
```

```
Node *q,*p;
```

```
q=L->first; (2分)
```

```
if (q==NULL)
```

```
return;
```

```
p=q->link;
```

```
while(p)
```

```
{
```

```
if (p->element==q->element) (2分)
```

```
{
```

```
q->link=p->link;
```

```
free(p);
```

```
p=q->link;
```

```
}
```

```
else (2分)
```

```
{
```

```
q=p;
```

```
p=p->link;
```

```
}
```

```
}
```

```
return;
```

```
}
```

参考答案 第3页共3页

# 《数据结构》

院(系、专业) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

请考生注意:

1. 答案请写在答题纸上, 写在试卷纸上一律无效!
2. 考试完毕, 请将答题纸和试卷纸交给监考老师, 不得带出考场!

得分

## 一、填空题(每空2分, 共计20分)

1. 数据结构的逻辑结构包括线性结构、(1)、图形结构和集合结构。
2. 后缀表达式  $ab+2^cd^*$  对应的中缀表达式为 (2)。
3. 两个集合元素的关键字为  $key_1$  和  $key_2$ , 给定散列函数  $H$ , 如果  $key_1 \neq key_2$  但是  $H(key_1) = H(key_2)$ , 则这种现象称为 (3)。
4. 将  $4 \times 5$  的二维数组  $A$  按行优先顺序存储到一维数组  $B$  中, 则  $B[14]$  存储的二维数组元素是 (4) ( $A[i][j]$  表示二维数组元素,  $i$  和  $j$  为行列下标, 数组  $A$ 、 $B$  下标均从 0 开始计数)。
5. 在有序表  $12, 14, 36, 51, 58, 70, 85, 92$  上查找 55, 若执行顺序搜索至少需要比较 (5) 次查找失败; 若执行对半搜索, 需要比较 (6) 次查找失败。
6. 利用一维数组  $A$  实现循环队列, 数组长度为 10, 当  $front=7$ ,  $rear=2$  时, 队列中元素个数为 (7), 最多再连续入队 (8) 个元素时队列满。
7. 若 4 阶  $B$ -树上有 25 个失败结点, 则该  $B$ -树上每个结点中关键字个数最少为 (9),  $B$ -树上关键字总数为 (10)。

得分

## 二、选择题(每小题2分, 共计20分)

1. 以下程序段的算法渐近时间复杂度为 ( )。  

```
void Func(int n) { for(int i = 1; i < n; i++) { i *= 2; printf("%i\n", i); } }
```

A、 $O(\log_2 n)$       B、 $O(n \log_2 n)$       C、 $O(\sqrt{n})$       D、 $O(n)$
2. 已知一棵二叉树结点的先序遍历序列为:  $B, A, F, E, D, C$ , 中序遍历序列为  $A, F, B, D, C, E$ , 则二叉树的后序遍历序列是 ( )。  
A、 $E, D, A, C, F, B$     B、 $E, D, C, A, F, B$     C、 $F, A, C, D, E, B$     D、 $F, D, C, A, E, B$
3. 包含 287 个结点的完全二叉树的高度是 ( )。  
A、8      B、9      C、10      D、11
4. 一棵二叉树中叶结点个数为 21, 度为 1 的结点个数为 25, 度为 2 的结点的个数为 ( )。  
A、20      B、22      C、24      D、26
5. 关于堆, 以下说法错误的是 ( )。  
A、序列  $23, 35, 38, 75, 69, 96$  是最小堆      B、序列  $64, 55, 39, 42, 27, 6$  是最大堆  
C、序列  $55, 56, 54, 52, 67, 31$  不是最大堆      D、序列  $89, 36, 32, 8, 86, 11$  是最大堆



- 6、设有5×5的数组A，其每个元素占2个字节，已知A[3][1]在内存中的地址是132（数组下标从0开始计数），按行优先顺序存储，A[2][3]的地址是（ ）。
- A、124                      B、126                      C、128                      D、130
- 7、堆栈和队列的主要区别是（ ）。
- A、逻辑结构不同                      B、存储结构不同  
C、限定元素插入和删除的位置不同                      D、运算结构不同
- 8、以下哪些算法是稳定的（ ）。
- A、冒泡排序和合并排序                      B、合并排序和简单选择排序  
C、直接插入排序和堆排序                      D、堆排序和冒泡排序
- 9、向空二叉平衡树依次插入关键字为82, 69, 24, 94的元素后，二叉平衡树根结点的关键字是（ ）。
- A、82                      B、69                      C、24                      D、94
- 10、关于散列表，以下说法正确的是（ ）。
- A、除留余数法可以解决冲突                      B、选择合适的散列函数可以避免冲突  
C、散列函数越复杂发生冲突概率越小                      D、散列表中元素存储位置与关键字值相关

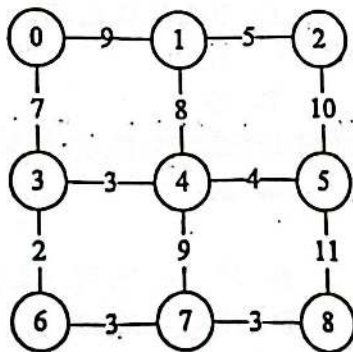
得分

三、简答题（每题8分，共计40分）

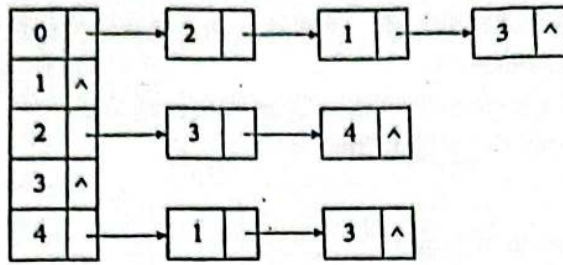
- 1、已知英文字母 A,B,C,D,E,F,G,H 及其对应权值 21,24,22,29,4,1,6,16，请给出以上英文字母的哈夫曼编码，要求该编码对应的哈夫曼树上左分支编码为0，右分支编码为1，且任意结点的左孩子权值不大于右孩子权值。
- 2、给定一个长度为11的散列表ht如下所示，采用双散列法解决冲突，两个散列函数分别为： $h_1(\text{key})=\text{key}\%11$ ， $h_2(\text{key})=\text{key}\%9+1$ 。请向散列表依次插入关键字为42,91,64,53的集合元素，给出插入完成后的散列表。

i	0	1	2	3	4	5	6	7	8	9	10
ht[i]		12	24	31			97			9	

- 3、请用普里姆算法以顶点2为起点构造下图的最小代价生成树，要求画出最小代价生成树的构造过程。



4、给定有向图的邻接表存储结构如下图所示，请给出从顶点 0 出发可以得到的深度优先遍历序列和宽度优先遍历序列（按照教材给定程序执行）。



5、对序列 68, 42, 61, 86, 38, 97, 20, 76 升序排序，请给出下列算法的第一趟排序结果：

(1) 直接插入排序：\_\_\_\_\_。

(2) 简单选择排序：\_\_\_\_\_。

(3) 两路合并排序：\_\_\_\_\_。

(4) 快速排序：\_\_\_\_\_。

得分

#### 四、算法填空题（每空2分，共计10分）

将Entry类型数据元素存储在List类型的线性表中，Entry与List类型定义如下：

```

typedef struct entry{
    KeyType key;
    DataType data;
}Entry;

typedef struct list{
    int n;
    Entry D[MaxSize];
}List;
  
```

请完成冒泡排序算法。

void BubbleSort(List \*list)

```

{ int i, j; /* i 标识每趟排序范围最后一个元素下标*/
  for( (1) ; i>0; i--)
  {
    BOOL isSwap = FALSE; /*标记一趟排序中是否发生了元素交换*/
    for(j=0; (2) ; j++)
    {
      if(list->D[j].key > list->D[j+1].key)
      {
        Swap(list->D, j, (3) ); /*交换操作*/
        (4) ;
      }
    }
    if( (5) ) break; /*如果本趟排序没有发生元素交换，排序完成*/
  }
}
  
```

得分

### 五、编程题 (每题5分, 共计10分)

设二叉树以二叉链表方式存储, 设二叉树结点和二叉树结构体定义如下:

```
typedef struct btnode {
    ElemType element; // ElemType 为可比较类型
    struct btnode* lchild, *rchild;
}BTNode;

typedef struct binarytree{
    BTNode* root;
}BinaryTree;
```

试编写递归程序, 实现:

(1) 查找二叉树中最大的数据元素

提示: 通过设计如下两个函数实现, 其中一个为递归函数; 如果最大值不唯一, 仅要求返回一个即可; 不用考虑树为空的情况。

```
ElemType MaxofBT(BinaryTree Bt)
```

```
ElemType Max(BTNode *p)
```

(2) 求二叉树的高度

提示: 通过设计如下两个函数实现, 其中一个为递归函数。

```
int DepthofBT(BinaryTree Bt)
```

```
int Depth(BTNode *p)
```

# 《数据结构》

院（系、专业）\_\_\_\_\_ 班级\_\_\_\_\_ 学号\_\_\_\_\_ 姓名\_\_\_\_\_

题号	一	二	三	四	五	总分
得分						

请考生注意：

1. 答案请写在答题纸上，写在试卷上一律无效。
2. 考试完毕，请将答题纸和试卷交给监考老师，不得带出考场。

得分	一、填空题 (每空2分, 共计20分)

(1) 树形/树形 结构/树	(2) $(a+b)^2+c*d$	(3) 冲突	(4) $A[2][4]$	(5) 5
(6) 3	(7) 5	(8) 4	(9) 1	(10) 24

得分	二、选择题 (每题2分, 共计20分)

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

得分

三、简答题 (每题8分, 共计40分)

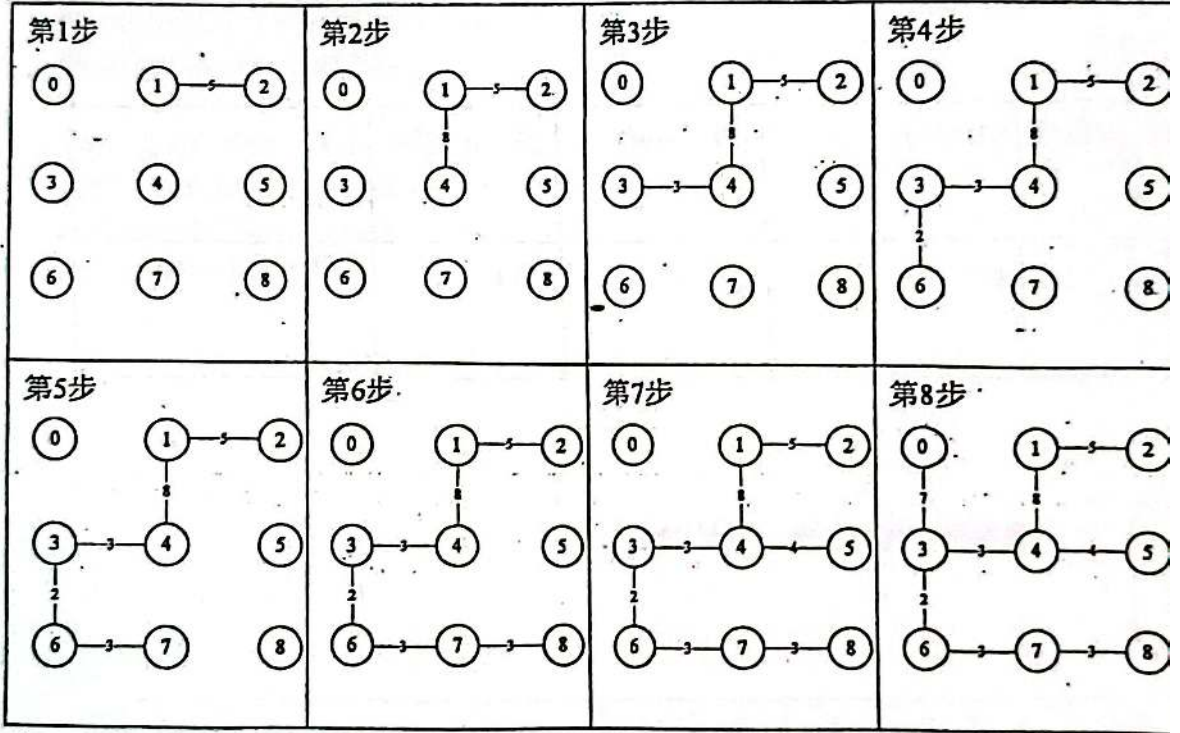
1. 一个编码1分

字母	哈夫曼编码	字母	哈夫曼编码
A	110	E	01001
B	00	F	01000
C	111	G	0101
D	10	H	011

2. 一个关键字2分

i	0	1	2	3	4	5	6	7	8	9	10
ht[i]	64	12	24	31		42	97	91		9	53

3. 最小代价生成树构造过程:



每步1分

4. 深度优先遍历序列: 0,2,3,4,1

宽度优先遍历序列: 0,2,1,3,4

5. 一题2分

(1) 直接插入排序: 42, 68, 61, 86, 38, 97, 20, 76

(2) 简单选择排序: 20, 42, 61, 86, 38, 97, 68, 76

(3) 两路合并排序: 42, 68, 61, 86, 38, 97, 20, 76

(4) 快速排序: 38, 42, 61, 20, 68, 97, 86, 76

得分

#### 四、算法填空题 (每空2分, 共计10分)

1.

(1)  $i=list->n-1$

(2)  $j<i$

(3)  $j+1$

(4)  $isSwap = TRUE$

(5)  $lisSwap$  或  $isSwap==FALSE$

得分

#### 五、编程题 (共计10分)

(1) 求最大值

```
- ElemType Max(BTNode *p)
{
    ElemType lmax=rmax=MIN; //MIN 为预定义的一个最小值
    if (p->lchild==NULL&& p->rchild==NULL)
        return p->element; //1 分
    if (lp->lchild)
        lmax = Max(p->lchild); //1 分
    if (lp->rchild)
        rmax = Max(p->rchild); //1 分
    ElemType chmax = lmax; //1 分, 比较左右子树最大值 和 p 结点值, 返回三者最
        //大值
    if (rmax > chmax)
```

```

        chmax=rmax;
        if(chmax>p->element)
            return chmax;
        return p->element;
    }
ElemType MaxofBT(BinaryTree Bt) {
    return Max(Bt.root); //1 分
}

```

(2) 求二叉树的高度

```

int Depth(BTNode *p) {
    int lh, rh;
    if (!p) return 0;
    lh = Depth(p->lchild); //1 分
    rh = Depth(p->rchild); //1 分
    if (lh > rh) return lh+1; //1 分
    return rh + 1; //1 分
}
int DepthofBT(BinaryTree Bt) {
    return Depth(Bt.root); //1 分
}

```

# 《 数据结构 》

院(系、专业) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

请考生注意:

1. 答案请写在答题纸上, 写在试卷纸上一律无效!

2. 考试完毕, 请将答题纸和试卷纸交给监考老师, 不得带出考场!

得分    一、填空题(每空 2 分, 共 20 分)

1. 在具有  $m$  个单元的循环队列中, 队满时共有个 \_\_\_\_\_ 数据元素。
2. 下列程序段中划线语句的渐近时间复杂度为 \_\_\_\_\_。
- ```
i=1; y=0;
do{
    y++; i=2*i;
}while(i<n)
```
3. 除二叉链表结构之外, 完全二叉树也可以采用 \_\_\_\_\_ 存储结构。
4. 二叉平衡树上的平衡因子有 \_\_\_\_\_ 种不同的值。
5. 后缀表达式:  $642+ / 32* +$  的值为 \_\_\_\_\_。
6. 二叉树的先序遍历序列为 ABDEGCF, 中序遍历序列为 DBGCEAF, 该二叉树根结点的左孩子结点是 \_\_\_\_\_。
7. 设有  $10 \times 10$  的整型数组 A, 其每个元素占 4 个字节, 已知  $A[0][0]$  在内存中的地址是 200, 按行优先顺序存储,  $A[4][6]$  的地址是 \_\_\_\_\_。
8. 设  $W = \{3, 2, 4, 5, 1\}$ , 以此权值集合构造的哈夫曼树的加权路径长度为 \_\_\_\_\_。
9. 高度为 5 的满二叉树共有 \_\_\_\_\_ 个分支结点。
10. 已知图的边集合  $E(G) = \{ \langle 0, 1 \rangle, \langle 1, 2 \rangle, \langle 4, 1 \rangle, \langle 4, 5 \rangle, \langle 5, 3 \rangle, \langle 2, 3 \rangle \}$ , 若采用邻接表存储, 则顶点 4 对应的边结点单链表中共有 \_\_\_\_\_ 个边结点。

得分    二、选择题(每小题 2 分, 共 20 分)

1. 在移动营业厅通过“取号、叫号”办理业务的服务模式符合 \_\_\_\_\_ 特征。
- A. 最小堆      B. 堆栈      C. 队列      D. 二叉树
2. 设某强连通图中有  $n$  个顶点, 则该强连通图最多有 \_\_\_\_\_ 边。
- A.  $n$       B.  $n*(n-1)$       C.  $n*(n-1)/2$       D.  $n*(n+1)$
3. 若入栈序列是 a, b, c, d, 则不可能得到的出栈序列为 \_\_\_\_\_。
- A. c, b, a, d      B. c, b, d, a      C. d, b, c, a      D. b, c, d, a



- 4、已知图的边集合  $E(G)=\{<6,1>,<1,2>,<4,1>,<4,5>,<5,3>,<2,3>\}$ ，则序列\_\_\_\_\_是该图的拓扑序列之一。
- A. 6, 3, 4, 5, 1, 2      B. 6, 1, 2, 3, 4, 5      C. 4, 5, 6, 1, 2, 3      D. 4, 3, 5, 2, 1, 6
- 5、下列选项中，\_\_\_\_\_是链接表结构不具有的特征。
- A. 可随机访问任一元素      B. 所需空间与线性长度成正比  
C. 不必事先估计存储空间      D. 插入、删除不需要移动元素
- 6、以下关键字序列\_\_\_\_\_是一个最小堆。
- A. 10, 72, 31, 25, 99, 58      B. 99, 25, 31, 72, 10, 58  
C. 10, 58, 25, 99, 31, 72      D. 10, 25, 58, 31, 99, 72
- 7、在下列排序算法中，\_\_\_\_\_的比较次数与元素的初始排列状态无关。
- A. 冒泡排序      B. 快速排序      C. 直接插入排序      D. 简单选择排序
- 8、利用哈夫曼算法构造出来的树是\_\_\_\_\_。
- A. 二叉排序树      B. 扩充二叉树      C. AVL 树      D. 完全二叉树
- 9、设 AVL 树中任一结点的子树为  $t_1$  和  $t_2$ ，则  $t_1$  和  $t_2$  的高度不可能为\_\_\_\_\_。
- A. 2 和 0      B. 0 和 1      C. 3 和 4      D. 10 和 11
- 10、在不带头结点的单链表中，当满足\_\_\_\_\_时，单链表为空。
- A.  $first = NULL$       B.  $first \rightarrow link = NULL$   
C.  $first \rightarrow link = first$       D.  $first \neq NULL$

|    |
|----|
| 得分 |
|    |

三、简答题（每题 7 分，共 42 分）

1、试将图 1 所示的二叉树转换成森林。

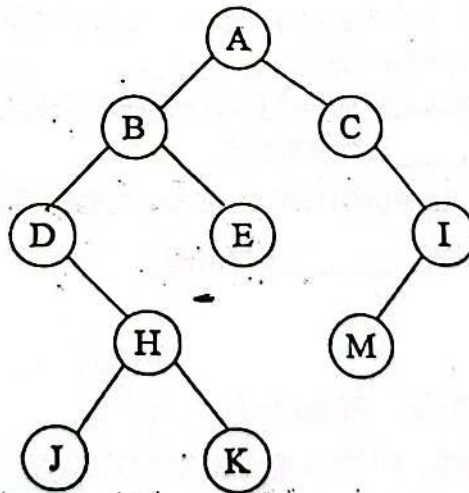


图 1

2. 设  $ht$  是长度为 13 的散列表, 采用二次探查法解决冲突, 散列函数为:  $h(key) = key \% 13$ , 试用关键字值序列 63, 24, 27, 59, 39, 88, 49 建立散列表。

|    |    |   |   |   |   |   |   |   |   |   |    |    |    |
|----|----|---|---|---|---|---|---|---|---|---|----|----|----|
|    | 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| ht | 78 |   |   |   |   |   |   |   |   |   |    |    |    |

3. 已知关键字序列为 12, 18, 27, 35, 56, 80, 试依照此顺序插入一棵空的 3 阶 B-树, 请给出完成所有元素插入之后的 B-树; 然后删除其中的关键字 12, 请给出删除 12 之后的 B-树。

4. 对于下图 2 所示的带权无向图, 请给出其对应的邻接矩阵。

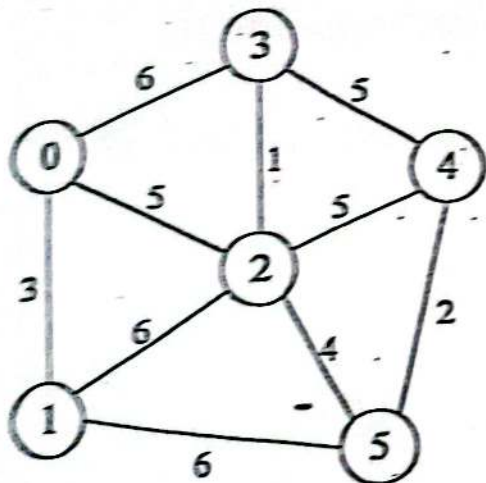


图 2

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 |   |   |   |   |   |   |
| 1 |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |
| 4 |   |   |   |   |   |   |
| 5 |   |   |   |   |   |   |

试用普里姆算法构造上图 2 所示的无向图的一棵最小代价生成树, 起点为 0 (要求出构造过程)。

对元素序列 49, 38, 66, 82, 13, 53, 3 按教材中冒泡排序算法和快速排序算法进行排序。

- (1) 写出第一趟冒泡排序的结果;
- (2) 写出第一趟快速排序的结果。

|    |
|----|
| 得分 |
|    |

#### 四、算法填空题（每空2分，共8分）

试完善如下算法程序，实现在顺序表的位置  $i$  处插入数据  $x$ ，顺序表结构体如

```
typedef struct {
    int n; //顺序表中元素的数量
    int maxLength; //顺序表的最大容量
    int *element; //用于存储数据的数组首地址
} SeqList;
```

请完成如下算法填空：

```
Status Insert(SeqList *L, int i, int x)
{
    int j;
    if ( (1) ) //判断下标 i 是否越界
        return ERROR;
    if ( (2) ) //判断顺序表存储空间是否已满
        return ERROR;
    for (j = L->n-1; j > i; j--)
        (3) ;
    L->element[i+1] = x;
    (4) ;
    return OK;
}
```

|    |
|----|
| 得分 |
|    |

#### 五、算法设计题（本题 10 分）

设二叉搜索树以二叉链表结构表示，结点结构体为：

```
typedef struct {
    int element;
    BSTNode *lchild;
    BSTNode *rchild;
} BSTNode;
```

试设计算法输出二叉搜索树中的最大值和最小值，要求算法的时间复杂度为  $O(\log_2 n)$ 。

```
void PrintMaxMin(BSTNode *r) //r 为树根结点
{
    ...
}
```

## 《 数据结构 》

### 一、填空题 (每空 2 分, 共 20 分)

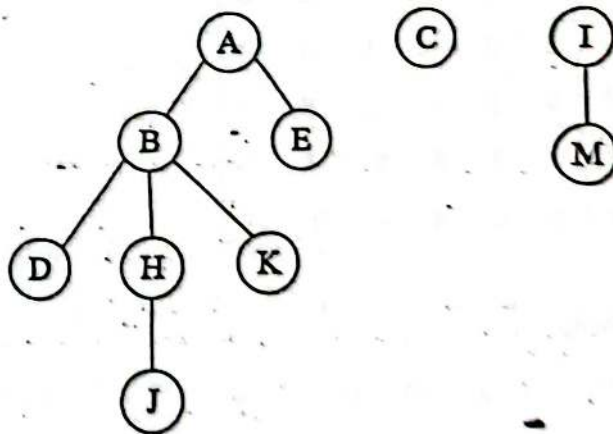
|     |               |           |   |   |   |     |    |    |    |
|-----|---------------|-----------|---|---|---|-----|----|----|----|
| 1   | 2             | 3         | 4 | 5 | 6 | 7   | 8  | 9  | 10 |
| m-1 | $O(\log_2 n)$ | 数组/<br>顺序 | 3 | 7 | B | 384 | 33 | 15 | 2  |

### 二、选择题 (每小题 2 分, 共 20 分)

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

### 三、简答题 (每题 7 分, 共 42 分)

1.



[评分] 第一棵树占 5 分, 第二、三棵树各占 1 分。

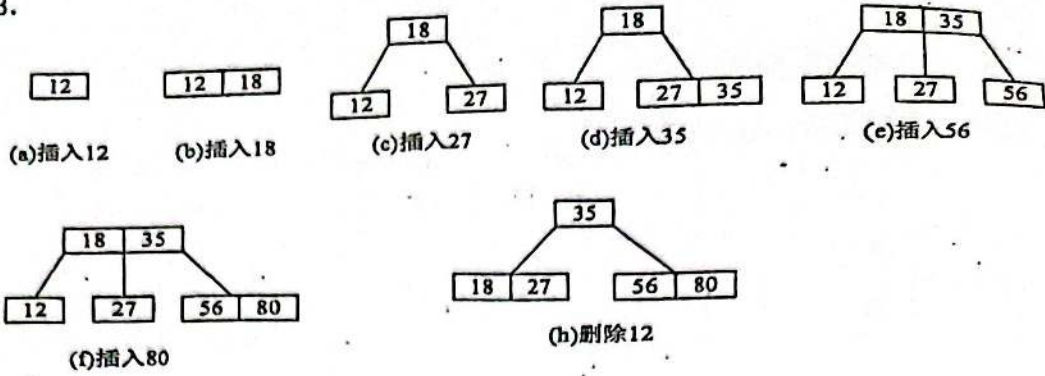
2.

|    |    |    |   |   |    |   |   |    |   |    |    |    |    |
|----|----|----|---|---|----|---|---|----|---|----|----|----|----|
|    | 0  | 1  | 2 | 3 | 4  | 5 | 6 | 7  | 8 | 9  | 10 | 11 | 12 |
| ht | 78 | 27 |   |   | 39 |   |   | 59 |   | 49 | 88 | 03 | 24 |

[评分] 每个元素 1 分。

1

3.



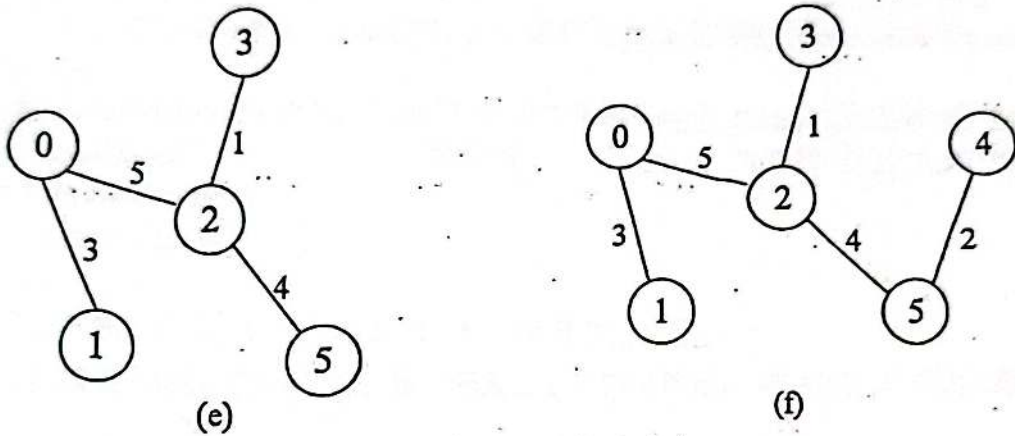
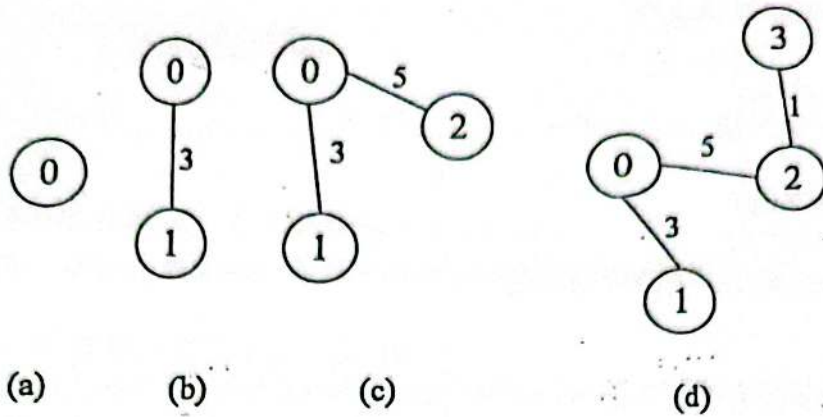
[评分] (f)全对也得 4 分, (g)全对得 3 分。若没有(f)和(g), 但有中间步骤(a)-(e), 每个正确步骤给 1 分, 但最多不超过 3 分。

4.

|   |    |    |    |    |    |    |
|---|----|----|----|----|----|----|
|   | 0  | 1  | 2  | 3  | 4  | 5  |
| 0 | -1 | 3  | 5  | 6  | 0  | 0  |
| 1 | 3  | -1 | 6  | 0  | 0  | 6  |
| 2 | 5  | 6  | -1 | 1  | 5  | 4  |
| 3 | 6  | 0  | 1  | -1 | 5  | 0  |
| 4 | 0  | 0  | 5  | 5  | -1 | 2  |
| 5 | 0  | 6  | 4  | 0  | 2  | -1 |

[评分] 错一个扣一分, 直至扣完。

5.



[评分] (b)~(e)五个步骤每个1分, 步骤(f)3分

6.

(4分) 第一趟冒泡排序的结果: 38 49 66 13 53 3 82

(3分) 第一趟快速排序的结果: 13 38 3 49 82 53 66

#### 四、算法填空题(每空2分, 共8分)

- (1)  $i < -1 \parallel i > L \rightarrow n - 1$  (注意:  $\parallel$ 左右交换位置都是正确的)
- (2)  $L \rightarrow n = L \rightarrow \text{maxLength}$
- (3)  $L \rightarrow \text{element}[j+1] = L \rightarrow \text{element}[j];$
- (4)  $L \rightarrow n = L \rightarrow n + 1;$  //  $L \rightarrow n++$ 也是正确的

五、请按照程序说明编写程序 (本题 10 分)

```
void PrintMaxMin(BSTNode *r)
```

```
{  
    int min, max;  
    BSTNode *p;  
    p=r;  
    while(p->lChild)           } 4分  
        p=p->lChild;  
    min = p->element; //获得最小值  
  
    p=r;  
    while(p->rChild)           } 4分  
        p=p->rChild;  
    max = p->element; //获得最大值  
  
    printf("最小值是 %d\n", min);           } 2分  
    printf("最大值是 %d\n", max);  
}
```

# 《数据结构》

院(系、专业) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

请考生注意:

1. 答案请写在答题纸上, 写在试卷纸上一律无效!
2. 考试完毕, 请将答题纸和试卷纸交给监考老师, 不得带出考场!

|    |
|----|
| 得分 |
|    |

## 一、填空题(每空 2 分, 共 20 分)

1. 若某算法的关键步骤执行次数为  $f(n) = 50 + 50n \log_2 n + 50n$ , 则其渐近时间复杂度为  $O(\quad)$ 。
2. 当线性表中查找运算很多时, 为了提高线性表的运算效率, 一般选择 \_\_\_\_\_ 存储表示法。
3. 若循环队列结构体定义如下, 队列满时队列中元素个数是 \_\_\_\_\_。  

```
typedef struct queue{  
    int front, rear, maxSize;  
    ElemType *element;  
} Queue;
```
4. 后缀表达式:  $8 \ 4 \ 4 \ + \ / \ 3 \ 2 \ * \ +$  的值为 \_\_\_\_\_。
5. 二维数组  $A[10][20]$  首地址 200, 每个元素占 1 个存储单元, 则  $A[6][12]$  的地址是 \_\_\_\_\_。
6. 一棵二叉树中度为 2 的结点个数为 1, 度为 1 的结点个数为 2, 则该树叶子结点个数为 \_\_\_\_\_。
7. 对半搜索是一种效率较高的搜索方法, 它要求线性表是有序表并采用 \_\_\_\_\_ 存储结构。
8. 一棵二叉平衡树的先序遍历序列是 5, 2, 3, 8, 10, 则其中序遍历序列是 \_\_\_\_\_。
9. 有一棵 5 阶 B-树高度为 3, 根结点的度至少为 \_\_\_\_\_。
10. 在散列表中, 不同关键字经过散列函数映射到相同地址的现象称为 \_\_\_\_\_。

|    |
|----|
| 得分 |
|    |

## 二、选择题(每小题 2 分, 共 20 分)

1. 假设某个循环队列的头指针为 front, 尾指针为 rear, 队列长度为 maxSize, 则判断队列为空的条件为 \_\_\_\_\_。  
A、 $front = rear$     B、 $front = rear - 1$     C、 $(rear + 1) \% maxSize = front$     D、 $front = rear = 0$



- 2、假设算法程序代码如下所示，则该算法的渐近时间复杂度为\_\_\_\_\_。
- ```

i=1; x=0;
do{ x++; i=3*i;} while(i<n);

```
- A、 $O(\log n)$       B、 $O(\log_3 n)$       C、 $O(n^2)$       D、 $O(n^3)$
- 3、具有  $n$  个顶点的强连通图中，边的条数最多为\_\_\_\_\_。
- A、 $n$       B、 $n^2$       C、 $n(n-1)/2$       D、 $n(n-1)$
- 4、对顺序表进行操作，以下说法错误的是\_\_\_\_\_。
- A、插入和删除元素的平均渐近时间复杂度相同  
B、查找成功的平均渐近时间复杂度为  $O(n)$   
C、数据元素在表中有序排列  
D、查找失败的渐近时间复杂度为  $O(n)$
- 5、给定一个 4 阶对称矩阵，约定以行优先规则存储下三角元素，则矩阵元素  $a_{23}$  在一维存储空间的下标  $k$  等于\_\_\_\_\_。
- A、5      B、6      C、7      D、8
- 6、在单链表中，向  $q$  指针指向的结点后插入一个  $p$  指针指向的新结点，应执行操作\_\_\_\_\_。
- A、 $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q;$   
B、 $p \rightarrow \text{link} = q \rightarrow \text{link}; q = p;$   
C、 $q \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = q;$   
D、 $p \rightarrow \text{link} = q \rightarrow \text{link}; q \rightarrow \text{link} = p;$
- 7、在下列序列中，\_\_\_\_\_不是最大堆。
- A、4,3,1,2      B、4,2,3,1      C、4,2,1,3      D、4,3,2,1
- 8、关于拓扑排序算法，以下说法错误的是\_\_\_\_\_ ( $n$  为顶点数)。
- A、当图以邻接矩阵方式存储时，其渐近时间复杂度为  $O(n^2)$   
B、该算法可用于判断有向图是否有回路  
C、当图以邻接表方式存储时，其渐近时间复杂度为  $O(n^2)$   
D、该算法只有对有向无环图才能输出正确的拓扑序列
- 9、关于关键路径算法，以下说法错误的是\_\_\_\_\_。
- A、该算法仅可作用于有向无环图  
B、关键路径是图中起点至终点的最短路径长度  
C、关键路径可用于计算一个工程的最短工期  
D、关键路径是图中起点至终点的最长路径长度
- 10、关于 Dijkstra 算法，以下说法错误的是\_\_\_\_\_ ( $n$  为顶点个数)。
- A、该算法执行 1 次就可以获得任意两顶点之间的最短路径  
B、该算法需要执行  $n$  次才能获得任意两顶点之间的最短路径  
C、该算法执行 1 次可以获得从指定顶点到其他  $n-1$  个顶点的最短路径  
D、当以邻接矩阵存储图时，该算法时间复杂度为  $O(n^2)$

得分

### 三、简答题（每题8分，共40分）

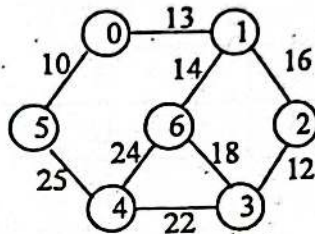
1、有长度为11的散列表，请采用二次探测法对依次输入的关键字12, 7, 48, 66建立散列表，散列函数  $H(key) = key \% 11$ 。

0	1	2	3	4	5	6	7	8	9	10
55	23	89	46							

2、向空的AVL树中，依次插入关键字2、60、7、55、9，画出AVL树的构建过程。

3、已知信号的权值集合为{4, 5, 6, 7, 10, 12, 17}，请构造哈夫曼树并计算该哈夫曼树的加权路径长度WPL。

4、用普里姆(Prim)算法，以0为源点，构造下图的最小代价生成树。要求画出各步的结果，并且计算生成树的代价。



5、利用快速排序算法对序列65、78、21、30、80、7、79、57、35、26进行升序排序，请写出前两趟排序结果。

得分

### 四、算法填空题（每空2分，共10分）

使用顺序表存储结构实现简单选择排序，请完成下列算法。

```
int FindMin(List list, int startIndex) {
    int i, minIndex = startIndex;
    for(i=startIndex+1; i < list.n; i++)
        { if( (1) ) (2) ; }
    return minIndex; }
```

```
void Swap(Entry* D, int i, int j)
{   if(i == j) return;
    Entry temp = *(D + i);
    *(D + i) = *(D + j);
    *(D + j) = temp; }
```

```
void SelectSort(List* list)
{   int minIndex, startIndex = 0;
    while( (3) )
```

```

{   minIndex = (4);
    Swap(list->D, startIndex, minIndex);
    (5);
}

```

得分

### 五、算法设计题（本题 10 分）

请设计实现以下算法，计算邻接表表示的图中任意顶点v的入度。

```

int Degree(Graph g, int v)
{
    .....
}

```

提示:

typedef struct enode //定义邻接表中边结点的结构体

```
{
```

```
    int adjVex;
```

```
    struct enode* nextArc;
```

```
} ENode;
```

typedef struct graph //定义图的结构体

```
{
```

```
    int n;
```

```
    ENode** A;
```

```
} Graph;
```

# 《数据结构》

## 一、填空题（每空 2 分，共 20 分）

1	2	3	4	5	6	7	8	9	10
$n \log_2 n$	顺序	maxSize-1	7	332	2	顺序	2,3,5,8,10	2	冲突

## 二、选择题（每小题 2 分，共 20 分）

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

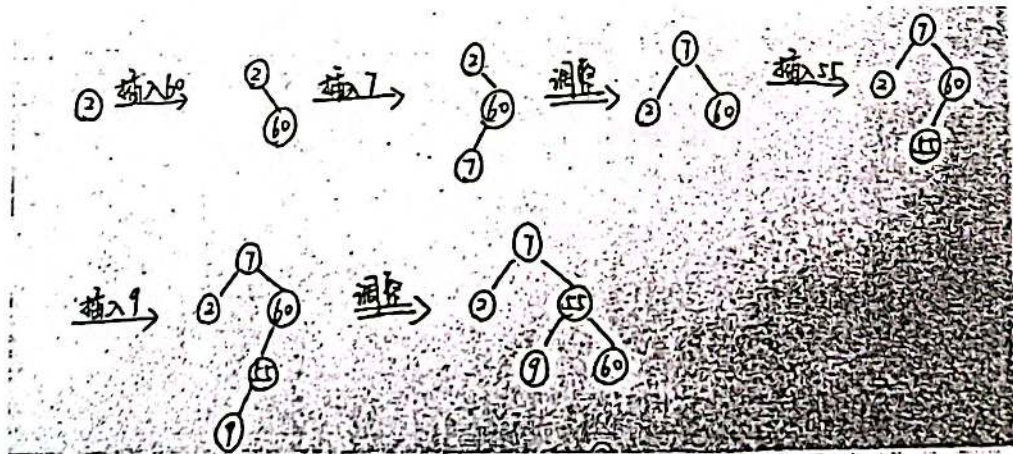
## 三、简答题（每题 8 分，共 40 分）

1.

0	1	2	3	4	5	6	7	8	9	10
55	23	89	46	48	12		7			66

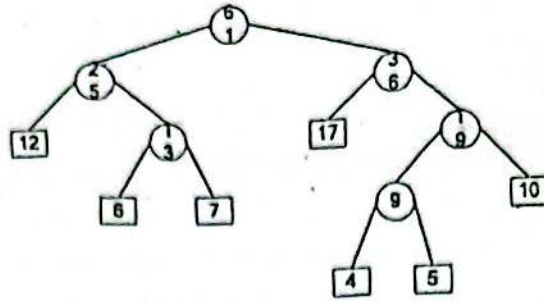
[评分] 每个空格 2 分，共 8 分。

2.



[评分] 前 4 步骤一步 1.5 分，最后一步 2 分

3.

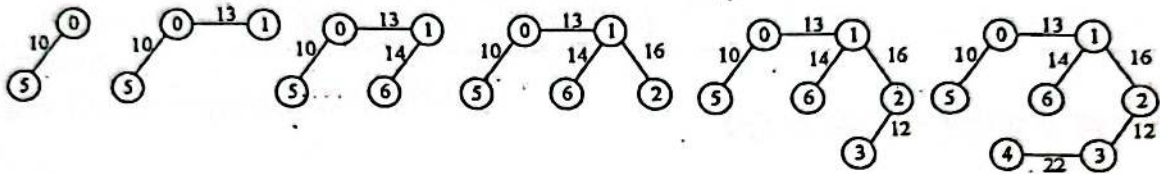


$$WPL = (4+5) \times 4 + (6+7+10) \times 3 + (12+17) \times 2 = 163$$

[评分] 哈夫曼树型全对给 6 分, 错一处扣 1 分, 扣完为止; WPL 占 2 分

4.

以 0 为起点, 依次为:



此生成树的代价为 87.

[评分] 共 6 个步骤, 对一个步骤得 1 分; 代价计算对得 2 分.

5.

第一趟: 57 26 21 30 35 7 65 79 80 78

第二趟: 7 26 21 30 35 57 65 79 80 78

[评分] 每趟 4 分

#### 四、算法填空题(每空 2 分, 共 10 分)

- (1) `list.D[i].key < list.D[minIndex].key`
- (2) `minIndex = i`
- (3) `startIndex < list->n-1`
- (4) `FindMin(*list, startIndex)`
- (5) `startIndex++`

## 五、算法设计题（本题 10 分）

```
int Degree(Graph g, int v)
{
    if ((v < 0) || (v > g.n - 1)) return -1; //2 分
    int indegree = 0; //1 分
    for (int i = 0; i < g.n; i++) //2 分
    {
        ENode *p = g.a[i]; //2 分
        while (p) //2 分
        {
            if (p->adjVex == v)
                indegree++;
            p = p->nextarc;
        }
    }
    return indegree; //1 分
}
```

# 《数据结构》期末

本试卷共 4 页；考试时间 110 分钟；

专业 \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

## 一、填空题 (20 分, 共 10 题)

1. 数据结构主要研究数据的 \_\_\_\_\_ 结构, 数据的存储结构以及在数据上执行的运算。
2. 设顺序表长度为 100, 若下标从 0 开始计, 则删除元素  $a_{10}$  需要移动 \_\_\_\_\_ 个元素。
3. 一棵二叉树中, 若叶结点的个数为 2011, 则度为 2 的结点个数为 \_\_\_\_\_。
4. 有向图进行拓扑排序时, 没有输出图中所有顶点, 说明图中存在 \_\_\_\_\_。
5. 线性表采用二分搜索必须满足两个条件: 线性表关键字必须是 \_\_\_\_\_; 存储结构必须采用顺序存储结构。
6. 二叉搜索树的 \_\_\_\_\_ 序遍历序列是一个按关键字递增排列的有序序列。
7. 设有一组记录的关键字为 {19, 14, 1, 69, 20, 27, 55, 79}, 散列函数为  $h(\text{key}) = \text{key} \% 11$ , 散列函数值为 3 的有 \_\_\_\_\_ 个。
8. 快速排序算法平均情况下的渐近时间复杂度为  $O(\quad)$ 。
9. 采用二次探查法解决冲突可能产生 \_\_\_\_\_ 聚集。
10. 图常见的两种存储结构有邻接矩阵和 \_\_\_\_\_。

## 二、选择题 (20 分, 共 10 题)

1. 一个算法必须在执行有穷步之后结束。这是算法的 \_\_\_\_\_。  
A. 有穷性      B. 正确性      C. 确定性      D. 可行性
2. 在指针  $p$  所指示的结点之后插入新结点  $s$  的操作是 \_\_\_\_\_。  
A.  $s \rightarrow \text{link} = p; p \rightarrow \text{link} = s;$       B.  $s \rightarrow \text{link} = p \rightarrow \text{link}; p \rightarrow \text{link} = s;$   
C.  $s \rightarrow \text{link} = p \rightarrow \text{link}; p = s;$       D.  $p \rightarrow \text{link} = s; s \rightarrow \text{link} = p;$
3. 栈和队列的共同点是 \_\_\_\_\_。  
A. 都是先进后出      B. 都是先进先出  
C. 只允许在端点处插入和删除元素      D. 没有共同点
4. 后缀表达式:  $5\ 3\ 2\ * \ 3 + 3 / +$  的值为 \_\_\_\_\_。  
A. 18      B. 7      C. 9      D. 8

5. 高度为5的二叉树至多有\_\_\_\_\_个结点。  
 A. 5                      B. 10                      C. 31                      D. 32
6. 采用对半查找方法查找长度为n的线性表时, 时间复杂度为\_\_\_\_\_。  
 A.  $O(n)$                       B.  $O(\log_2 n)$                       C.  $O(n)$                       D.  $O(\log_2 n)$
7. n个顶点的无向图采用邻接矩阵表示, 则该矩阵的大小是\_\_\_\_\_。  
 A. n                      B.  $(n-1)^2$                       C.  $n^2$                       D.  $n-1$
8. 一个无向连通图的生成树是一个\_\_\_\_\_连通子图。  
 A. 极大                      B. 极小                      C. 有时极大                      D. 有时候极小
9. 下列排序方法中, 排序过程中的比较次数与排序方法无关的是\_\_\_\_\_。  
 A. 简单选择排序法                      B. 直接插入排序法  
 C. 快速排序法                      D. 冒泡排序法
10. 散列表的长度为11, 下标范围是[0, 10], 散列函数为 $h(key) = key \% 11$ 。采用线性探查法解决冲突, 依次将关键字7, 38, 5, 16插入空的散列表中。则关键字16在散列表中存放的下标是\_\_\_\_\_。  
 A. 5                      B. 6                      C. 7                      D. 8

### 三、简答题 (30分, 共5题)

1. 有二叉树如图1所示, 写出该二叉树的前序遍历序列和中序遍历序列。

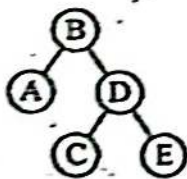


图1

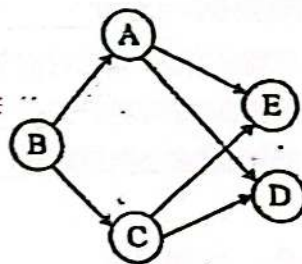


图2

2. 写出图2所有可能的拓扑排序。
3. 设有向图的邻接表表示如图3所示, 请给出每个顶点的入度。

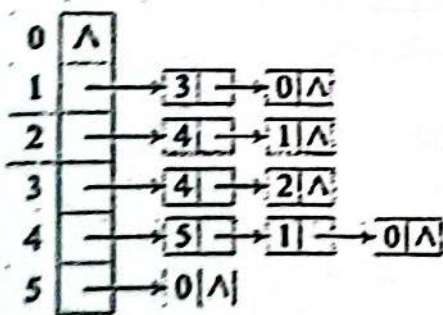


图3



4. 空二叉搜索树中依次插入 33, 44, 99, 22, 11, 55, 画出最终所构建二叉搜索树。
5. 设  $W = \{5, 6, 7, 8, 9\}$ ,
  - (1) 画出由权值集合  $W$  构造的哈夫曼树。
  - (2) 计算加权路径长度。

#### 四、判断题 (10 分, 共 5 题)

1. 线性结构只能用顺序结构来存放, 非线性结构只能用非顺序结构来存放。
2. 简单选择排序是稳定的排序算法。
3. 散列函数越复杂越好, 因为这样随机性好, 冲突概率小。
4. 完全二叉树一定存在度为 1 的结点。
5. 在一非空二叉树的中序遍历中, 根结点的右边是其右子树上的所有结点。

#### 五、程序填空题 (10 分, 共 1 题)

1. 以下程序是对半搜索的迭代实现, 请填写完整。

BOOL BSearch2(List lst, KeyType k, T \*x).

```

{
    int mid, low=(1)_____, high = lst.Size-1;
    while ((2)_____)
    {
        mid=(low+high)/2;
        if (k<lst.Elements[mid].Key) high = (3)_____;
        else if ((4)_____) low = mid+1;
        else {
            *x=lst.Elements[mid]; return TRUE;
        }
    }
    (5)_____
}

```

六、编程题 (10分, 共1题)

1. 用二叉链表方式存储二叉树。试编写函数Count1, 求一棵二叉树的结点总数。并编写Count接口函数, 让其调用Count1函数。

```
typedef int K;
typedef struct btnode{
    K Element;
    struct btnode* LChild, *RChild;
}BTNode;
typedef struct btree{
    struct btnode* Root;
}BTree;
```

《数据结构》期末答案

一、填空题 (20分, 共10题)

1	2	3	4	5	6	7	8	9	10
逻辑	89	2010	有向回路	有序	中	2	nlogn	二次	邻接表

二、选择题 (20分, 共10题)

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

三、简答题 (30分, 共5题)

1.

前序遍历序列: BADCE (3分)

自觉遵守考试规则, 诚信考试, 绝不作弊

中序遍历序列: ABCDE (3分)

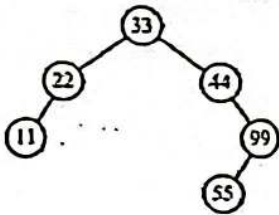
2. 每个1分, 全对再加2分

BACDE BACED BCAED BCAED

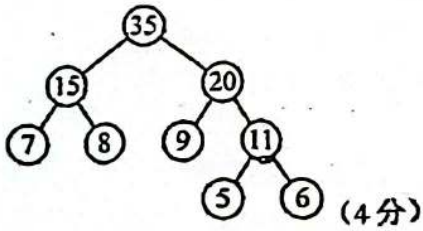
3. 每个1分

顶点	入度
0	3
1	2
2	1
3	1
4	2
5	1

4.



5.



WPL = (5+6) \* 3 + (7+8+9) \* 2 = 33 + 48 = 81 (2分)

#### 四、判断题 (10分, 每题2分)

1	2	3	4	5
×	×	×	×	√

#### 五、程序填空题 (10分, 每空2分)

(1) 0

(2) low <= high

(3) mid-1

(4) k>lst.Elements[mid].Key

(5) return FALSE;

## 六、编程题 (10分, 共1题)

1.

```
int Count(BTree Bt){  
    return Count1(Bt.Root);  
}
```

```
int Count1(BTNode* p){  
    if(!p) return 0;  
    else return Count1(p->LChild)+ Count1(p->RChild) + 1;  
}
```

# 《 数据结构 》

本试卷共 4 页； 考试时间 110 分钟；

专业 \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

装  
订  
线  
内  
不  
要  
答  
题

## 一、填空题 (20 分, 共 10 题)

1. 数据结构主要研究数据的 逻辑 结构, 数据的存储结构以及在数据上执行的运算。
2. 设顺序表 A 长度为 100, 若下标从 0 开始计, 则删除元素 A[10] 需要移动 \_\_\_\_\_ 个元素。
3. 一棵二叉树中, 若叶结点的个数为 2011; 则度为 2 的结点个数为  $\frac{2010}{2} = 1005$
4. 无向图的连通分量是其 \_\_\_\_\_ 连通子图。
5. 线性表采用二分搜索必须满足两个条件: 线性表关键字必须是 \_\_\_\_\_; 存储结构必须采用顺序存储结构。
6. 二叉搜索树的 \_\_\_\_\_ 序遍历序列是一个按关键字递增排列的有序序列。
7. 设有一组记录的关键字为 {19, 14, 1, 69, 20, 27, 55, 79}, 散列函数为  $h(key) = key \% 11$ , 散列函数值为 3 的有 \_\_\_\_\_ 个。
8. 快速排序算法平均情况下的渐近时间复杂度为  $O(\quad)$ 。
9. 采用二次探查法解决冲突可能产生 \_\_\_\_\_ 聚集。
10. 图常见的两种存储结构有邻接矩阵和 \_\_\_\_\_。

## 二、选择题 (20 分, 共 10 题)

1. 一个算法必须在执行有限步之后结束。这是算法的 \_\_\_\_\_。  
A. 有穷性      B. 正确性      C. 确定性      D. 可行性
2. 在指针 p 所指示的结点之后插入新结点 s 的操作是 \_\_\_\_\_。  
A.  $s \rightarrow link = p; p \rightarrow link = s;$       B.  $s \rightarrow link = p \rightarrow link; p \rightarrow link = s;$   
C.  $s \rightarrow link = p \rightarrow link; p = s;$       D.  $p \rightarrow link = s; s \rightarrow link = p;$
3. 栈和队列的共同点是 \_\_\_\_\_。  
A. 都是先进后出      B. 都是先进先出  
C. 只允许在端点处插入和删除元素      D. 没有共同点
4. 后缀表达式:  $532 * 3 + 3 / +$  的值为 \_\_\_\_\_。  
A. 18      B. 7      C. 9      D. 8

47

5. 高度为 5 的二叉树至多有 \_\_\_\_\_ 个结点。  
 A. 5                      B. 10                      C. 31                      D. 32
6. 采用对半查找方法查找长度为  $n$  的线性表时, 时间复杂度为 \_\_\_\_\_。  
 A.  $O(n^2)$                       B.  $O(n \log_2 n)$                       C.  $O(n)$                       D.  $O(\log_2 n)$
7.  $n$  个顶点的无向图采用邻接矩阵表示, 则该矩阵的大小是 \_\_\_\_\_。  
 A.  $n$                       B.  $(n-1)^2$                       C.  $n^2$                       D.  $n-1$
8. 一个无向连通图的生成树是一个 \_\_\_\_\_ 连通子图。  
 A. 极大                      B. 极小                      C. 有时极大                      D. 有时极小
9. 下列排序方法中, 排序过程中的比较次数与排序方法无关的是 \_\_\_\_\_。  
 A. 简单选择排序法                      B. 直接插入排序法  
 C. 快速排序法                      D. 冒泡排序法
10. 散列表的长度为 11, 下标范围是  $[0, 10]$ , 散列函数为  $h(key) = key \% 11$ . 采用线性探查法解决冲突. 依次将关键字 7, 38, 5, 16 插入空的散列表中. 则关键字 16 在散列表中存放的下标是 \_\_\_\_\_。  
 A. 5                      B. 6                      C. 7                      D. 8

三、简答题 (30 分, 共 5 题)

1. 有二叉树如图 1 所示, 写出该二叉树的先序遍历序列和中序遍历序列。

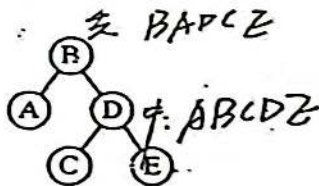


图 1

21	0	0	0	0
0	0	11	0	0
0	1	0	3	8
0	0	0	2	0

图 2

2. 写出图 2 中稀疏矩阵的行三元组表示及快速转置算法中  $K[col]$  数组中各元素值。
3. 设有向图的邻接表表示如图 3 所示, 请给出每个顶点的入度。

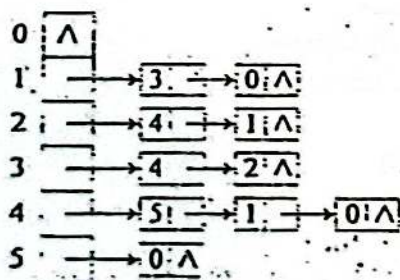


图 3

4. 空二叉搜索树中依次插入 33, 44, 99, 22, 11, 55. 画出最终所构建二叉搜索树。

5. 设  $W = \{5, 6, 7, 8, 9\}$ , 要求左子树根节点的权值小于等于右子树根节点权值。
- (1) 画出由权值集合  $W$  构造的哈夫曼树。
  - (2) 计算加权路径长度。

四、判断题 (10分, 共5题, 对的记“√”, 错的记“×”)

1. 线性结构只能用顺序结构来存放, 非线性结构只能用非顺序结构来存放。
2. 简单选择排序是稳定的排序算法。
3. 散列函数越复杂越好, 因为这样随机性好, 冲突概率小。
4. 完全二叉树一定存在度为1的结点。
5. 在一非空二叉树的中序遍历中, 根结点的右边是其右子树上的所有结点。

五、程序填空题 (10分, 共1题)

1. 以下程序是对半搜索的迭代实现, 请填写完整。

```

BOOL BSearch2(List lst, KeyType k, T *x)
{
    int mid, low=(1)_____, high = lst.Size-1;
    while ((2)_____)
    {
        mid=(low+high)/2;
        if (k<lst.Elements[mid].Key) high = (3)_____;
        else if ((4)_____) low = mid+1;
        else {
            *x=lst.Elements[mid]; return TRUE;
        }
    }
    (5)_____
}

```

六、编程题 (10 分, 共 1 题)

1. 用二叉链表方式存储二叉树。试编写函数Countl, 求一棵二叉树的结点总数。并编写Count接口函数, 让其调用Countl函数。

```
typedef int K;
typedef struct btnode{
    K Element;
    struct btnode* LChild, *RChild;
}BTNode;
typedef struct btree{
    struct btnode* Root;
}BTree;
```



## 《 数据结构 》 答案

### 一、填空题 (20分, 共10题)

1	2	3	4	5	6	7	8	9	10
逻辑	89	2010	有向回路	有序	中	2	$n \log_2 n$	二次	邻接表

### 二、选择题 (20分, 共10题)

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

### 三、简答题 (30分, 共5题)

1.  
 前序遍历序列: BADCE (3分)  
 中序遍历序列: ABCDE (3分)

2.

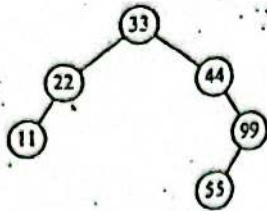
$$\begin{pmatrix} 0 & 0 & 21 \\ 1 & 2 & 11 \\ 2 & 1 & 1 \\ 2 & 3 & 3 \\ 2 & 4 & 8 \\ 3 & 3 & 2 \end{pmatrix}$$

col	0	1	2	3	4
K[col]	0	1	2	3	5

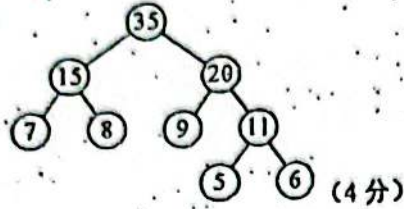
3. 每个1分

顶点	入度
0	3
1	2
2	1
3	1
4	2
5	1

4.



5.



(4分)

$$WPL = (5+6) * 3 + (7+8+9) * 2 = 33 + 48 = 81 \quad (2分)$$

#### 四、判断题 (10分, 每题2分)

1	2	3	4	5
×	×	×	×	√

#### 五、程序填空题 (10分, 每空2分)

(1) 0

(2) low <= high

(3) mid-1

(4) k > lst.Elements[mid].Key

(5) return FALSE;

#### 六、编程题 (10分, 共1题)

1:

```

int Count(BTree Bt){
    return Count1(Bt.Root);
}
  
```

```

int Count1(BTNode* p){
    if(!p) return 0;
    else return Count1(p->LChild) + Count1(p->RChild) + 1;
}
  
```

# 《 数据结构 》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	总分
得分							

注意事项:

- (1) 全部试题解答都必须写在答卷上。
- (2) 试卷和答卷均需写上学号姓名。考试结束，将试卷和答卷一起交上。

得分

一、 判断题 (每小题 2 分, 共 10 分) (请回答“√”或“×”)

- (X) 1. 线性表顺序存储的优点是存储密度大, 且插入和删除运算效率高。
- (√) 2. 对二叉搜索树进行中序遍历, 得到的结点序列按升序排列。
- (√) 3. 若一棵二叉树根的右子树为空树, 则其对应的森林只有一棵树。
- (√) 4. B-树是一种适合外存存储的数据结构。
- (X) 5. 求单源最短路径的迪杰斯特拉算法通过按边上权值的从小到大次序, 逐一考察图中“边”的方式来产生最短路径。

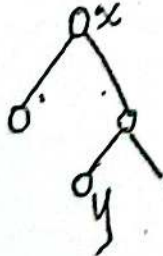
得分

二、 选择题 (每小题 2 分, 共 10 分)

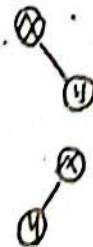
1. 设单链表的结点有两个域: element 和 link, 在指针 p 所指示的结点之后插入指针 q 所指示的新结点的操作是 B。
- A.  $q \rightarrow link = p; p \rightarrow link = q;$       B.  $q \rightarrow link = p \rightarrow link; p \rightarrow link = q;$   
 C.  $q \rightarrow link = p \rightarrow link; p = q;$       D.  $p \rightarrow link = q; q \rightarrow link = p;$

2. 对二叉树中的一个结点 x, 设其在先序遍历序列中的序号为  $pre(x)$ , 在后序序列中的序号为  $post(x)$ 。若树中结点 x 是结点 y 的祖先, 则下列结论中 \_\_\_\_\_ 是正确的。

- A.  $pre(x) > pre(y)$  和  $post(x) < post(y)$   
 B.  $pre(x) > pre(y)$  和  $post(x) > post(y)$



(数据结构 A) 试卷 第 1 页 共 4 页



- C.  $pre(x) < pre(y)$  和  $post(x) < post(y)$   
 D.  $pre(x) < pre(y)$  和  $post(x) > post(y)$

3. 采用对半搜索方法查找长度为  $n$  的有序表时, 查找每个元素时 平均比较次数应 A 对应的二叉判定树的高度 (假定高度大于等于 2).  
 A. 小于 B. 等于 C. 大于 D. 大于等于
4. 使用二次探查法构造散列表是为了避免 B.  
 A. 基本冲突 B. 基本聚集 C. 二级聚集 双列法 D. 二次冲突
5. 从未排序序列中挑选元素, 并将其依次放入已排序序列 (初始时空) 的一端, 这种排序方法称为 A.  
 A. 插入排序 B. 归并排序 C. 选择排序 D. 快速排序

得分

三、填空题 (每空 2 分, 共 12 分)

1. 设删除表中的每个元素的概率是相等的, 则在长度为  $n$  的顺序表中, 删除一个元素平均需要移动的元素个数的计算公式为  $\frac{1}{n} \sum_{i=1}^{n-1} (n-i+1)$
2. 表达式  $a * c - b / d^2$  的后缀形式是  $ac * bd^2 / -$
3. 设有三维数组  $A[4][5][3]$ , 每个元素占 1 个单元, 按 行 优先顺序存储 (最右下标变化最快), 数组存储区的起始地址为  $b$ , 则元素  $A[2][4][2]$  的存储位置是为  $b + 2 \times 5 \times 3 + 4 \times 3 + 2$
4. 堆中根结点的编号为 0, 编号为  $k$  的结点的右孩子的编号为  $2k+2$
5. 无向 图的邻接矩阵为对称矩阵.
6. 文件有下列四种基本的组织方式: 顺序文件 索引文件 链式文件 散列文件

插:  $\sum_{i=1}^{n-1} \frac{1}{n+1} (n-i)$

$b + \frac{2 \times 5 \times 3}{30} + \frac{4 \times 3}{12} + \frac{2}{2}$

34

$4 \times 5 \times 2$

得分

四、解答题 (每题 8 分, 共 48 分)

设模式串  $P = "aabaaba"$ , 设有主串  $S = "aabaabaabaaba"$

(1) 计算模式串的失败函数和改进的失败函数的值.

(2) 当 KMP 算法在主串的字符合 'x' 处失配时, 根据该字符的改进的失败函数的值, 下一趟应当分别由主串和模式串中的什么字符开始比较?

	0	1	2	3	4	5	6
P	a	a	b	a	a	b	a
fj	-1	0	1	0	1	0	1
fj'	-1	-1	1	-1	-1	1	-1

改进的失败函数

	0	1	2	3
P	a	a	b	a
fj'	-1	0	1	0
fj	-1	-1	1	-1

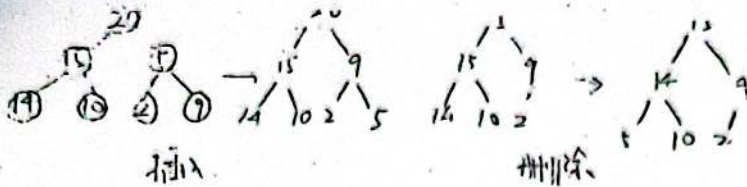
试第 2 页共 4 页

模式串  $m = 0$

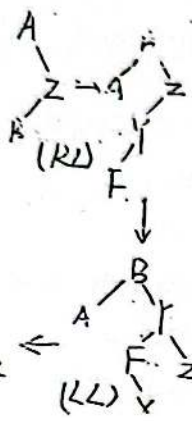
主串  $i =$

$f(2) = g(4)$

主串  $i =$



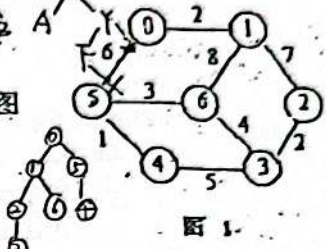
设有初始 max 堆组成的优先权队列为: 20, 15, 5, 14, 10, 2. 对其  
 (1) 先插入元素 9, 画出插入并调整后的堆 (即优先权队列).  
 (2) 再删除最大元素, 画出删除并调整后的堆 (即优先权队列).



3. 请从空树开始, 依次输入 A, Z, B, Y, F, X, 构造二叉平衡树. 画出构造二叉平衡树的过程. 如果需要重新平衡, 请指出应执行何种旋转.

4. 有无向图如图 1 所示  
 (1) 给出此图的邻接矩阵存储结构.  
 (2) 画出从顶点 0 开始的广度优先搜索该图的生成树.

0	0	2	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



5. 无向图如图 1 所示  
 (1) 从图 1 的顶点 0 出发, 用 Prim 算法构造一棵最小代价生成树 (画出中间过程).  
 (2) 计算该生成树的代价.

6. 使用快速排序算法对元素序列 (23, 43, 36, 30, 29, 54, 76, 28) 进行排序.  
 (1) 写出对上述序列进行第一趟排序后的结果;  
 (2) 给出快速排序的最坏情况的渐近时间复杂度表示;  
 (3) 试举出一种可以改善快速排序最坏情况时间性能的措施.

① 将  $A[\text{left} + \text{right} / 2]$  与  $A[\text{left}]$  交换  
 ② 将  $A[k]$  与  $A[\text{left}]$  交换  
 ③ 取  $A[\text{left}], A[\text{right}], A[\text{left} + \text{right} / 2]$  中的中间值与  $A[\text{left}]$  交换

算法填空题 (每空 2 分, 共 8 分)

1. 补充完整下列在顺序表中插入新元素的函数 Insert(i, x). x 插入的位置在元素 elements[i] 之后. 若 i=-1, 则将新元素 x 插在最前面. 若插入成功, 返回 true, 否则返回 false.

```

template<class T>
bool SeqList<T>::Insert(int i, T x)
{
    if (i < -1 || i > n-1 || n == maxLength) {
        cout << "Out Of Bounds" << endl; return false;
    }
    for (int j = n-1; j > i; j--) elements[j+1] = elements[j];
    elements[i+1] = x;
}
    
```

```
n++; return true;
```

2. 补充完整下列冒泡排序算法。函数 Swap(a, b)的功能是交换两个实在参数的值。

```
template <class T>
void BubbleSort(T A[], int n)
{
    int i, j, last;
    i = n - 1;
    while (i > 0) {
        last = 0;
        for (j = 0; j < i; j++)
            if (A[j] > A[j+1])
                Swap(A[j], A[j+1]);
                last = j;
        i = last;
    }
}
```

得分

### 六、 算法设计题 (12分)

在以二叉链表表示的二叉树类 BinaryTree 中增加一个公有成员函数 Degree2InTree() 和一个私有递归成员函数 Degree2InTree(BTNode<T>\* t)。前者调用后者，求二叉树类的对象中度为 2 的结点数目。请分别实现这两个函数。

函数原型为：

```
template <class T>
int BinaryTree<T>:: Degree2InTree( )
和
```

```
template <class T>
int BinaryTree<T>:: Degree2InTree(BTNode<T>* t);
```

```
template <class T>
int BinaryTree<T>:: Degree2InTree( )
{
    return Degree2InTree(root);
}
```

```
template <class T>
int BinaryTree<T>:: Degree2InTree(BTNode<T>* t)
{
    if (!t->lchild || !t->rchild) return 0;
    else return Degree2InTree(t->lchild) +
        Degree2InTree(t->rchild) + 1;
}
```

(数据结构A) 试卷 第 4 页 共 4 页

# 《数据结构》

(考试时间 100 分钟)

班级 \_\_\_\_\_ 姓名 \_\_\_\_\_ 学号 \_\_\_\_\_ 得分 \_\_\_\_\_

一、单项选择题 (本大题共 15 小题, 每小题 2 分, 共 30 分) 在每小题列出的四个选项中只有一个符合题目要求, 请将其代码填在题后的括号内, 错选或未选均无分。

1. 算法必须具备输入、输出和

[ C ]

- A. 计算方法                      B. 排序方法  
C. 解决问题的有限运算步骤      D. 程序设计方法

2. 有  $n$  个节点的顺序表中, 算法的时间复杂度是  $O(1)$  的操作是

[ A ]

- A. 访问第  $i$  个节点 ( $1 \leq i \leq n$ )  
B. 在第  $i$  个节点后插入一个新节点 ( $1 \leq i \leq n$ )  
C. 删除第  $i$  个节点 ( $1 \leq i \leq n$ )  
D. 将  $n$  个节点从小到大排序

3. 单链表的存储密度

[ C ]

- A. 大于 1                          B. 等于 1  
C. 小于 1                          D. 不能确定

4. 设将整数 1,2,3,4,5 依次进栈, 最后都出栈, 出栈可以在任何时刻 (只要栈不空) 进行, 则出栈序列不可能是

[ B ]

- A. 23415 ✓                      B. 54132  
C. 23145 ✓                      D. 15432

5. 循环队列 SQ 的存储空间是数组  $d[m]$ , 队头、队尾指针分别是  $front$  和  $rear$ , 则执行出队后其头指针  $front$  值是

[ D ]

- A.  $front=front+1$                   B.  $front=(front+1)\%(m-1)$   
C.  $front=(front-1)\%m$               D.  $front=(front+1)\%m$

6. 在一个具有  $n$  个结点的有序单链表中插入一个新结点并仍然保持有序的时间复杂度是

[ B ]

- A.  $O(1)$                       B.  $O(n)$                       C.  $O(n^2)$                       D.  $O(n \log n)$

7. 设二维数组  $A[0..m-1][0..n-1]$  按行优先顺序存储, 则元素  $A[i][j]$  的地址为

[ B ] 5

- A.  $LOC(A[0][0])+(i*m+j)$       B.  $LOC(A[0][0])+(i*n+j)$   
 C.  $LOC(A[0][0])+((i-1)*n+j-1)$       D.  $LOC(A[0][0])+((i-1)*m+j-1)$
8. 一个非空广义表的表头 [D ]  
 A. 一定是子表      B. 一定是原子  
 C. 不能是子表      D. 可以是原子, 也可以是子表
9. 具有  $n$  个节点的完全二叉树的深度为 [A ]  
 A.  $\lceil \log_2(n+1) \rceil - 1$       B.  $\log_2 n + 1$   
 C.  $\log_2 n$       D.  $\lfloor \log_2 n \rfloor$
10. 若要惟一地确定一棵二叉树, 只需知道该二叉树的 [D ]  
 A. 前序序列      B. 中序序列  
 C. 前序和后序序列      D. 中序和后序序列
11. 在一个无向图中, 所有顶点的度数之和等于图的边数的 \_\_\_\_\_ 倍 [C ]  
 A.  $1/2$       B. 1  
 C. 2      D. 4
12. 拓扑排序运算只能用于 [C ]  
 A. 带权有向图      B. 连通无向图  
 C. 有向无环图      D. 无向图
13. 在所有排序方法中, 关键字比较的次数与记录的初始排列次序无关的是 [D ]  
 A. 希尔排序      B. 冒泡排序  
 C. 插入排序      D. 选择排序
14. 下列排序算法中时间复杂度不受数据初始状态影响, 恒为  $O(n^2)$  的是 [C ]  
 A. 堆排序      B. 冒泡排序  
 C. 直接选择排序      D. 快速排序
15. 二分查找要求节点 [A ]  
 A. 有序、顺序存储      B. 有序、链接存储  
 C. 无序、顺序存储      D. 无序、链接存储



二、填空题 (本大题共 10 小题, 每小题 2 分, 共 20 分) 不写解答过程, 将正确的答案写在每小题的空格内。错填或不填均无分。

16. 数据的逻辑结构分为两大类, 它们是线性结构和 非线性结构。
17. 在单链表中 (假设结点指针域名称为 next), 删除指针 P 所指结点的后继结点的语句是 p->next=p->next->next。
18. 已知循环队列用数组 data[n] 存储元素值, 用 front, rear 分别作为头尾指针, 则当前元素个数为 (rear-front+n)%n。
19. 若 n 为主串长, m 为子串长, 则串的朴素匹配算法最坏的情况下需要比较字符的总次数为 (n-m+1) × m。
20. 广义表 ((a), ((b), j, ((d)))) 的表尾是 ((b), j, ((d)))。
21. 已知二叉树有 61 个叶子节点, 且仅有一个孩子的节点数为 45, 则总节点数为 166。
22. 解决计算机与打印机之间速度不匹配问题, 须要设置一个数据缓冲区, 应是一个 队列 结构。
23. n 个顶点 e 条边的图采用邻接表存储, 深度优先遍历算法的时间复杂度为 O(n+e)。
24. 对于 n 个关键字的集合进行冒泡排序, 在最坏情况下所需要的时间为 O(n<sup>2</sup>)。
25. 在一个长度为 n 的顺序表中的第 i 个元素 (1 ≤ i ≤ n) 之前插入一个元素时, 需向后移 n-i+1 个元素。

三、解答题 (本大题共 4 小题, 共 25 分)

26. 对于下面的稀疏矩阵, 画出其三元组法存储表示 (假设下标从 0 开始), (5 分)

$$\begin{pmatrix} 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -6 & 0 \\ 7 & 0 & 0 & 0 & 0 & 24 \\ 0 & 0 & 0 & 18 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 \end{pmatrix}$$

答案:

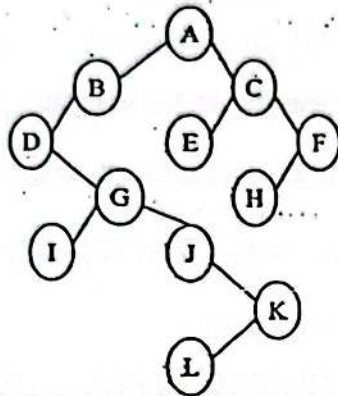
	行号	列号	值
0	0	2	14
1	1	4	-6
2	2	0	7
3	2	5	24
4	3	3	18
5	4	1	15

27. 已知一棵二叉树的中序序列和后序序列分别如下, 请画出该二叉树。(5分)

中序序列: D I G J L K B A E C H F

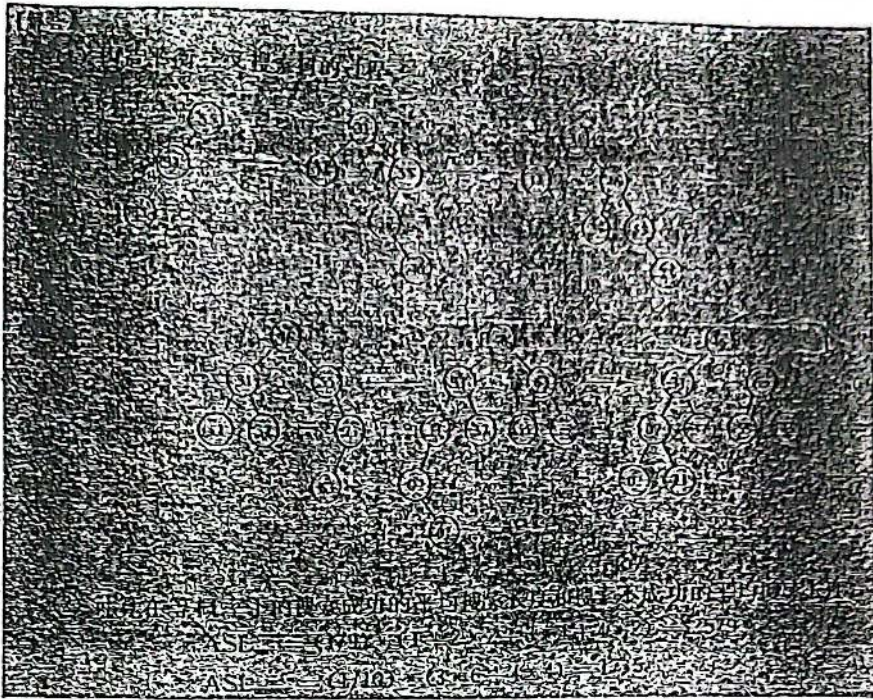
后序序列: I L K J G D B E H F C A

答案:



28. 设有一个关键码的输入序列 {55,31,11,37,46,73,63,02,07}, (7分)

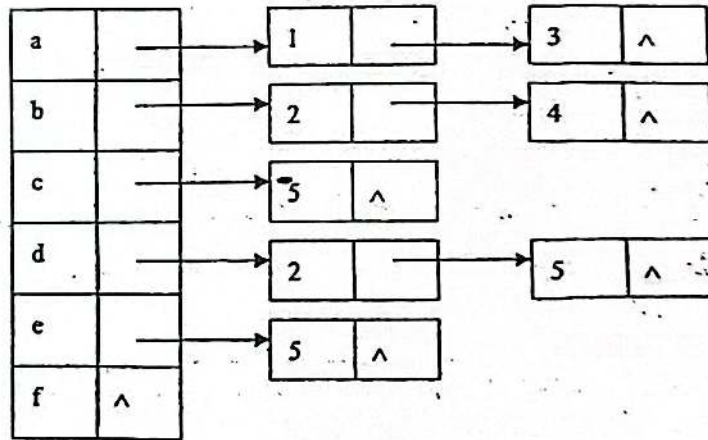
- (1) 从空树开始构造平衡二叉搜索树, 画出每加入一个新结点时二叉树的形态。若发生不平衡, 指明需做的平衡旋转的类型及平衡旋转的结果。(3分)
- (2) 计算该平衡二叉搜索树在等概率下的搜索成功的平均搜索长度和搜索不成功的平均搜索长度。(4分)



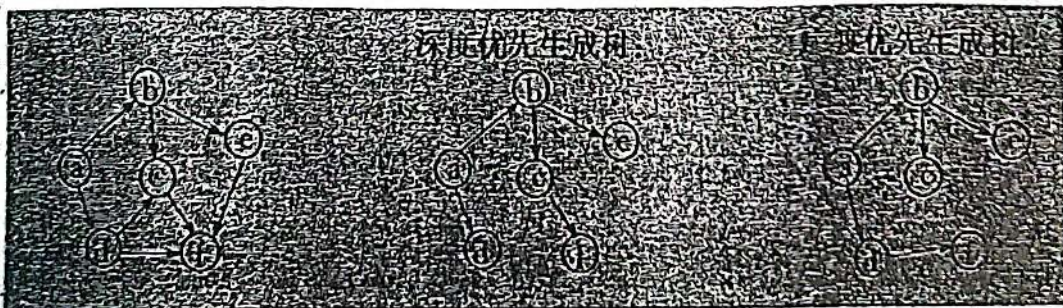
29. 已知一个图的邻接表如下所示。(8分)

(1) 画出该图的图形;(4分)

(2) 根据邻接表分别画出从顶点 a 出发进行深度优先和广度优先遍历所生成的生成树。(4分)



答案:



#### 四、 算法阅读题 (本大题共 3 小题, 每小题 5 分, 共 15 分)

30. 设线性表的  $n$  个结点定义为  $(a_0, a_1, \dots, a_{n-1})$ , 在顺序表上实现的插入和删除算法如下, 请在空白处填入适当内容。(顺序表的最大可容纳项数为  $MaxSize$ )

```
Template <class Type> int SeqList<Type>::Insert(Type &x, int i) {  
    If (i<0 || i>last+1 || last== (1) ) return 0;  
    Else {  
        Last++;  
        For(int j=last;j>i;j--) data[j]= (2) ;  
        (3) ;  
        Return 1;  
    }  
}
```

```
Template <class Type> int seqList<Type>::Remove(Type &x){  
    int i=Find(x);  
    if(i>=0) {  
        last-;  
        for (int j= (4) ;j<=last;j++) data[j]= (5) ;  
        return 1;  
    }  
    return 0;  
}
```

答案:

- (1)  $MaxSize-1$
- (2)  $data[j-1]$
- (3)  $Data[j]=x$
- (4)  $i$
- (5)  $data[j+1]$

31. 阅读下面的算法, 请回答下列问题:

- (1) 试说明算法的功能。
- (2) 当执行该程序时, 输入 12345678-1, 输出什么结果?

```
#define StackSize 200  
typedef int DataType;  
typedef struct {  
    DataType data[StackSize];  
    int top;  
} SeqStack;  
void Push(SeqStack *s, DataType x)  
{  
    if(s->top!=StackSize-1)  
        s->data[++s->top]=x;  
}  
DataType Pop(SeqStack *s)  
{
```

```

        if(s->top!=1)
            return s->data[s->top-];
    }
    void main()
    {
        DataType i;
        SeqStack s;
        s.top=1;
        scanf("%d",&i);
        while(i!=-1)
        {
            push(&s,i);
            scanf("%d",&i);
        }
        while(s->top!=1)
        {
            i=Pop(&s);
            printf("%6d",i);
        }
    }
}

```

答案:

(1)程序的功能是把输入的一串整数(用-1 做结束标记) , 按照与输入相反的次序输出用栈实现这一功能。

(2) 输出结果 8 7 6 5 4 3 2 1.

32. 已知二叉树的存储结构为二叉链表, 阅读下面算法. 说明该算法的功能.

```

Template<class Type>
int BinaryTree<Type>::height(BinTreeNode<Type> *t){
    if(t==NULL) return -1;
    int hl=height(t->leftChild);
    int hr=height(t->rightChild);
    return 1+(hl>hr?hl:hr);
}

```

答案: 该算法的功能是统计二叉树的高度。

## 五、算法设计题（本题共 10 分）

33. 设一棵二叉树以二叉链表表示，试以成员函数形式编写有关二叉树的递归算法。

(1) 统计二叉树中度为 1 的结点个数。（5 分）

(2) 统计二叉树中度为 2 的结点个数。（5 分）

（提示：递归算法如 32 题所示）

解答：(1) 统计二叉树中度为 1 的结点个数。

```
Template<class Type>
Int BinaryTree<Type> ::Degree1(BinTreeNode<Type> *t) const {
    If(t==NULL) return 0;
    If(t->leftchild!=NULL && t->rightchild==NULL || t->leftchild==NULL && t->rightchild!=NULL)
        Return 1+Degree1(t->leftchild)+Degree1(t->rightchild);
    Return Degree1(t->leftchild)+Degree1(t->rightchild);
}
```

(2) 统计二叉树中度为 2 的结点个数。

```
Template<class Type>
Int BinaryTree<Type> ::Degree2(BinTreeNode<Type> *t) const {
    If(t==NULL) return 0;
    If(t->leftchild!=NULL && t->rightchild!=NULL)
        Return 1+Degree2(t->leftchild)+Degree2(t->rightchild);
    Return Degree2(t->leftchild)+Degree2(t->rightchild);
}
```

## 《 数据结构 》

院(系) \_\_\_\_\_ 班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

题号	一	二	三	四	五	六	七	八	总分
得分									

### 填空题

写出表达式  $a*b+c/d$  的后缀形式\_\_\_\_\_。

【答案】  $a b * c d / +$

已知一无向图  $G=(V,E)$ ，其中  $V=\{a, b, c, d, e\}$ ， $E=\{(a, b), (a, d), (a, c), (d, c), (b, e)\}$ ，现用某一种遍历方法从顶点  $a$  开始遍历图，得到的序列为  $abecd$ ，则采用的是\_\_\_\_\_遍历方法。

【答案】深度优先

在顺序表长度为  $n$  中，平均在表中插入一个元素需要移动元素的个数可用计算公式为\_\_\_\_\_。

【答案】  $E_i = \sum_{i=1}^{n-1} \frac{1}{n+1} (n-i-1) = \frac{n}{2}$

一个表长为  $n$  的线性表，其排序时间最快为\_\_\_\_\_。

【答案】 $O(n)$

### 选择题

具有  $n$  个顶点的有向完全图中，边的总数为 ( ) 条。

A)  $n(n+1)$

B)  $n(n-1)$

C)  $n(n-1)/2$

D)  $n(n+1)/2$

【答案】B

设一个栈输入序列是 1、2、3、4、5，则下列序列中不可能是栈的输出序列是 ( )。

A) 32541

B) 15432

C) 14523

D) 23145

【答案】C 只能是14532

二叉树的前序遍历为 EFHIGJK，中序遍历序列为 HFIEJKG。该二叉树根结点的右子树的根是 ( )

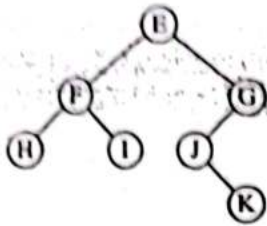
A) E

B) F

C) G

D) H

【答案】C



对有14个元素的有序表A[1]-A[14]作对半查找，查找元素A[4]时的被比较元素依次为（）

- A. A[1], A[2], A[3], A[4]                      B. A[7], A[3], A[5], A[4]  
C. A[1], A[2], A[7], A[4]                      D. A[7], A[5], A[3], A[4]

【答案】B

第1次：范围[1, 14]，中间元素是 $(1+14)/2 = 7$

第2次：范围[1, 6]，中间元素是 $(1+6)/2 = 3$

第3次：范围[4, 6]，中间元素是 $(4+6)/2 = 5$

第4次：范围[4, 4]，中间元素是 $(4+4)/2 = 4$

设有一个长度为100且已排好序的表，用对半搜索进行查找，若搜索不成功，则至少要比较\_\_\_\_\_次。

( )

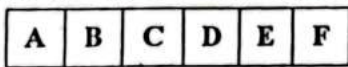
- A. 9                      B. 8                      C. 7                      D. 6

【答案】D

长度为100的有序表进行对半查找，查找失败时比较 $\lfloor \log_2 100 \rfloor$ 次或者 $\lfloor \log_2 100 \rfloor + 1$ 次，即6或7次。

简答题

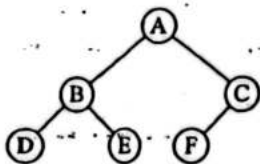
用一维数组存放的一棵完全二叉树如图所示：



图

写出前序、中序、后序遍历该二叉树时访问结点的顺序。

【答案】



前序遍历序列：ABDECF

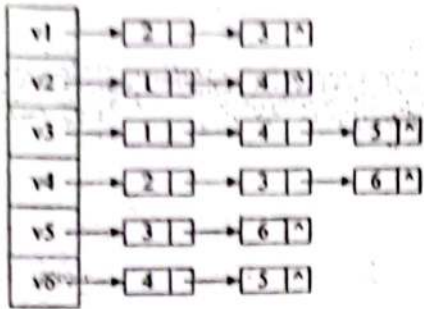
中序遍历序列：DBE AFC

后序遍历序列：DEBFCA

图的邻接表表示一个给定的无向图。

- (1) 给出从顶点 v1 开始，用深度优先搜索法进行遍历时的顶点序列；
- (2) 给出从顶点 v1 开始，用广度优先搜索法进行遍历时的顶点序列。





【答案】

深度优先遍历序列: v1, v2, v4, v3, v5, v6

广度优先遍历序列: v1, v2, v3, v4, v5, v6

解答题

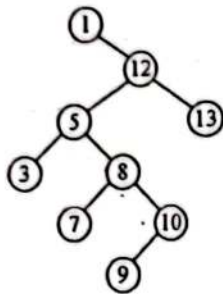
设数据集  $d = \{1, 12, 5, 8, 3, 10, 7, 13, 9\}$ , 试完成下列各题:

(1) 依次取  $d$  中各数据, 构造一棵二叉搜索树  $bt$ .

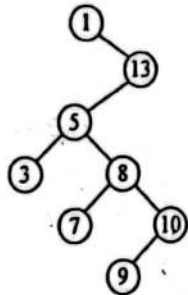
(2) 画出在二叉树  $bt$  中删除 12 后的树结构.

【答案】

(1)

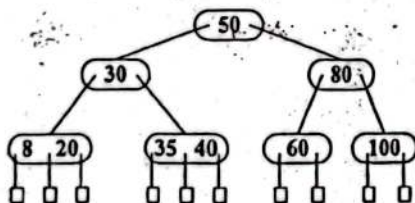


(2) 删除 12 后



对图的 3 阶 B-树, 依次执行下列操作, 画出各步操作的结果.

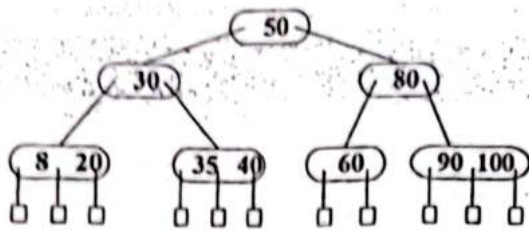
(1) 插入 90; (2) 插入 25; (3) 插入 45; (4) 删除 60;



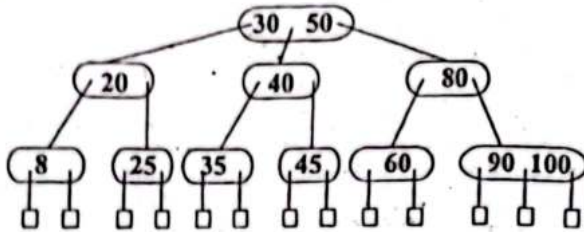
图

【答案】

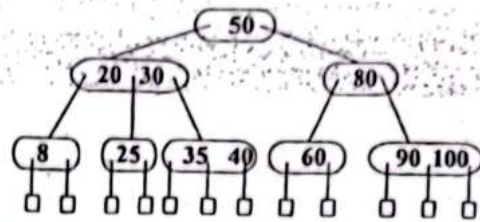
插入 90



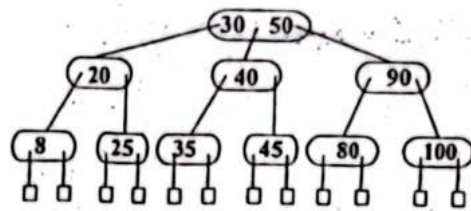
插入 45



插入 25



删除 60



### 程序阅读题

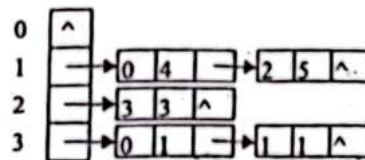
图采用邻接表存储表示，边结点的结构如图所示，下面的程序是邻接表类LinkedGraph的某个成员函数

```
template <class T>
void LinkedGraph<T>::A()
```

```
{
    int *in=new int[n];
    for (int i=0;i<n;i++) in[i]=0;
    ENode<T> *p;
    for (i=0;i<n;i++)
    {
        p=a[i];
        while (p)
        {
            in[p->adjvex]++;
            p=p->nextarc;
        }
    }
    cout<<endl;
    for (i=0;i<n;i++) cout<<i<<": "<<in[i]<<";"<<endl;
    delete []in;
}
```

adjvex	weight	nextarc
--------	--------	---------

图



图

- (1) 请说明该成员函数的作用是什么?
- (2) 若有一个邻接表如图 8 所示，请给出执行该函数的结果?

【答案】

- (1) 打印图中所有顶点的入度
- (2) 结果  
0 : 2  
1 : 1  
2 : 1  
3 : 1

### 算法题

在以二叉链表表示的二叉树类 BinaryTree 中增加一个成员函数 LeavesInTree()。该模板函数为递归函数，其功能是求二叉树类 BinaryTree 的对象中叶子结点的数目。实现该递归函数。函数原型如下：

```
template <class T>
int BinaryTree<T>::LeavesInTree()
```

【答案】

```
template <class T>
int BinaryTree<T>::LeavesInTree()
{
    return Leaf(root);
}
```

```
template <class T>
int BinaryTree<T>::Leaf(BTNode<T> *t)
{
    if (t == NULL) return 0;
    if ((t->lChild == NULL)&&(t->rChild == NULL)) return 1;
    return Leaf(t->lChild) + Leaf(t->rChild);
}
```

在不带头结点的单链表中删除一个关键字值为 x 的元素。函数原型如下：

```
template <class T>
bool SingleList<T>::Delete(T x)
```

【答案】

```
template <class T>
bool SingleList<T>::Delete(T x)
```

```
{
    BTNode<T> *p = first;
    BTNode<T> *q = NULL;
    while (p && p->data == x)
    {
        q = p;
        p = p->link;
    }
    if (!p) return false;
    if (q)
        {q->link = p->link;}
    else
        {first = p->link;}
    delete p;
    n--;
    return true;
}
```

# 《 数据结构 》

班级 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_ 得分 \_\_\_\_\_

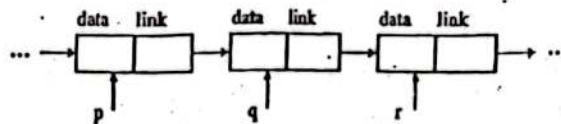
题号	一	二	三	四	五	六	七	八	九	十
得分										

## 一、填空题 (6小题, 每小题2分, 共12分)

1. 顺序表中各元素之间的地址是\_\_\_\_\_。
2. 若有n个表并存, 并且处理时各表的长度和表的总数都会动态变化, 则应采用\_\_\_\_\_存储结构。
3. 栈的TOP运算的功能是\_\_\_\_\_, 但此运算不删除栈顶元素。
4. 对树进行后序遍历时, 最后出现的结点, 在先序遍历中将\_\_\_\_\_出现。
5. AOV网络中活动之间的领先关系是一种\_\_\_\_\_, 它具有传递性和反自反性。
6. 若待排序序列为0 81 15 8 9 15 71, 经过某种排序后, 得到的序列为0 8 9 15 15 71 81, 则此排序算法的稳定性如何? 答案是\_\_\_\_\_。

## 二、单项选择题 (7小题, 每小题2分, 共14分)

1. 将q和r所指结点的先后位置交换且不出现断链现象, 以下错误的程序段是\_\_\_\_\_。

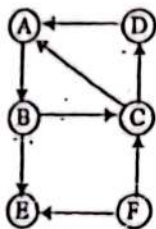


- (A)  $q \rightarrow \text{link} = r \rightarrow \text{link}; p \rightarrow \text{link} = r; r \rightarrow \text{link} = q;$
  - (B)  $r \rightarrow \text{link} = q; q \rightarrow \text{link} = r \rightarrow \text{link}; p \rightarrow \text{link} = r;$
  - (C)  $p \rightarrow \text{link} = r; q \rightarrow \text{link} = r \rightarrow \text{link}; r \rightarrow \text{link} = q;$
  - (D)  $q \rightarrow \text{link} = r \rightarrow \text{link}; r \rightarrow \text{link} = q; p \rightarrow \text{link} = r;$
2. WINDOWS操作系统中页面调度时, 先来先服务算法是\_\_\_\_\_的典型应用实例。  
 (A) 数组 (B) 堆栈 (C) 队列 (D) 二叉树
  3. 中缀表达式  $A + (B - C / D) * E$  的后缀形式是\_\_\_\_\_。  
 (A)  $ABCD / * E +$  (B)  $ABCD / - E * +$   
 (C)  $ABCD + * E /$  (D)  $ABCD + E - / *$
  4. 在对非空二叉树进行中序遍历的序列中, 根结点右边\_\_\_\_\_。  
 (A) 只有左子树上的部分结点 (B) 只有根的左子树上的全部结点或无结点  
 (C) 只有右子树上的部分结点 (D) 只有根的右子树上的全部结点或无结点

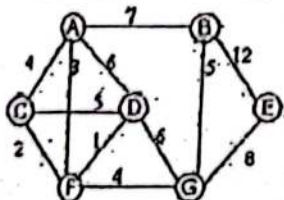
5. 有个非零元素的稀疏矩阵  $A_{m \times n}$ ，采用三元组表示，则快速转置的时间复杂度是\_\_\_\_\_。
- (A)  $O(m \times n)$  (B)  $O(m+n)$   
 (C)  $O(n+1)$  (D)  $O(n \times 1)$
6. 二叉搜索树中，最小元素结点的左子树\_\_\_\_\_，它的右子树\_\_\_\_\_。
- (A) 一定为空，不一定为空 (B) 不一定为空，一定不为空  
 (C) 一定不为空，不一定为空 (D) 不一定为空，不一定为空
7. 对非连通图进行广度优先搜索得到\_\_\_\_\_。
- (A) 树 (B) 森林  
 (C) 生成树 (D) 生成森林

三、简答题 (8小题，每小题6分，共48分)

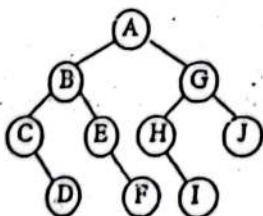
1. 输入序列为 (21 60 12 50 45 80)，请先建立二叉搜索树，再从此树上将60删除。
2. 设散列表的长度为 11，若采用双散列法解决冲突，试以散列函数  $h_1(key) = key \% 11$ ， $h_2(key) = key \% 9 + 1$ ，从空表开始，依次插入下列关键字值序列：81 25 80 35 60 45，建立散列表。请画出该散列表。
3. 有向图见下图。给出强连通分量的定义并画出强连通分量。



4. 使用普里姆 (Prim) 算法以A为源点，构造下图的最小代价生成树；画出各步的结果。

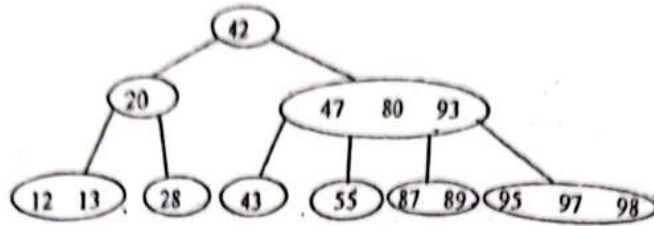


5. 画出下图中的二叉树所对应的森林；若 X 结点是其双亲 Y 的右孩子，则在对应的树或森林中 X 是 Y 的什么结点？



6. 当以边  $\langle 0, 1 \rangle$ ,  $\langle 1, 3 \rangle$ ,  $\langle 1, 2 \rangle$ ,  $\langle 2, 5 \rangle$ ,  $\langle 5, 0 \rangle$ ,  $\langle 4, 2 \rangle$ ,  $\langle 4, 3 \rangle$ ,  $\langle 2, 0 \rangle$  的次序从只有6个顶点没有边的图开始，通过插入这些边，建立邻接表。
- (1) 画出该邻接表；
- (2) 在所建立的邻接表上，进行以0为起始顶点的深度优先遍历，写出遍历结果。

7. 设字符集合S={A, B, C, D, E}, 各字符的使用频率为W={6, 7, 10, 6, 3}
- (1) 画出哈夫曼树; (生成新结点时, 新结点的左子树根的权值小于等于右子树根的权值)
- (2) 求该哈夫曼树的带权路径长度;
8. 设有四阶B-树, 如下图所示 (未画出失败结点), 画出插入关键字96后的B-树;



#### 四、程序填空题(每空2分, 共8分)

下面是快速排序程序, 请补充完整:

```

template <class T>
void QSort(T A[], int left, int right)
{ int i, j;
  if ( left<right )
  { i=left; j=right+1;
    do
      (do i++; while ( (1) );
      do j--; while ( (2) );
      if (i<j) Swap(A[i], A[j]);
    }while (i<j);
    Swap( (3) );
    QSort(A, left, j-1); QSort(A, (4) );
  }
}
  
```

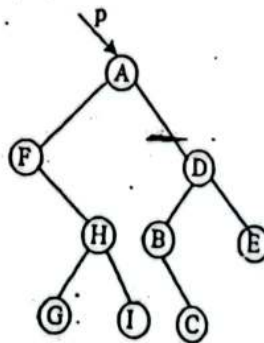
#### 五、程序阅读题(每个小题3分, 共6分)

算法程序如下:

```

.....
int k; //设k是Btree类中的数据成员, 并有初值0
.....
template <class T>
int BTree<T>::F(BTNode<T>* u)
{ if(u)
  { if(u->lchild==null && u->rchild==null) k++;
    F(u->lchild);
    F(u->rchild);
  }
}
  
```

- (1) 说明该程序的作用是什么?
- (2) 执行F(p)后, k的值是多少?



## 六、程序设计题(12分)

已知带头的单链表(SingleList)的结点(Node)有两个私有的数据成员, data 和 link, 其中 data 是结点关键字, link 是指向 Node 的指针. SingleList 中私有的数据成员有两个, first 和 length. 其中 first 是指向第一个结点的指针, length 是当前单链表中结点的个数. 请完成:

- (1) 写出结点 Node 和单链表 SingleList 的 C++ 类模板; (Node 只要求写出私有的数据成员, SingleList 不用从 LinearList 继承, 其成员函数只要列出(2)中的 Insert 函数即可)
- (2) 假设单链表中的结点是有序递增的, 设计成员函数 Insert(const T& x), 在单链表中插入元素 x 且保证链表的有序性. (单链表中任意两个结点的关键字都不相等)

一、填空题 (6小题, 每小题2分, 共12分)

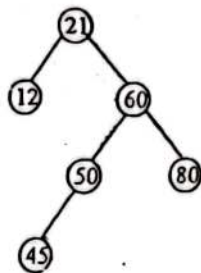
1. 连续的
2. 链式
3. 取得栈顶元素的值
4. 最先
5. 拟序
6. 不稳定

二、单项选择题 (7小题, 每小题2分, 共14分)

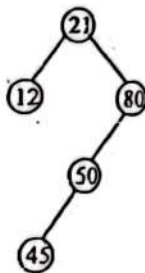
BCBDCAD

三、简答题 (8小题, 1每题6分, 共48分)

1. 4分



2分

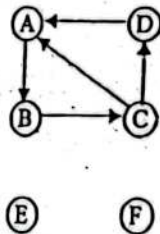


2. (每个1分)

0	1	2	3	4	5	6	7	8	9	10
	80	35	25	81	60	45				

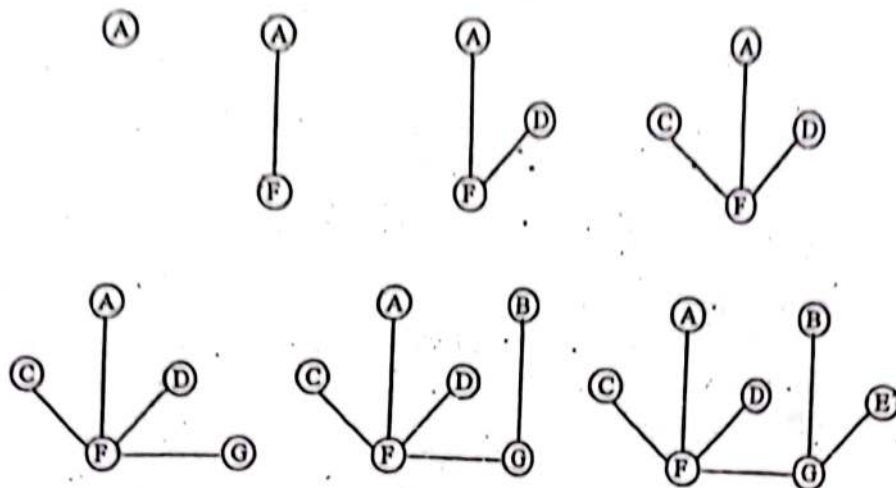
3. (1) 有向图的一个极大强连通子图称为该图的一个强连通分量。(2分)

(2) (4分)

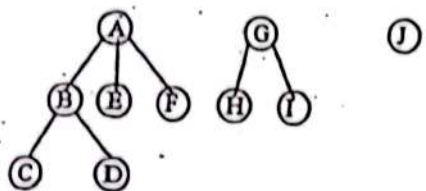




(每步1分)



5.            2分        1分        1分



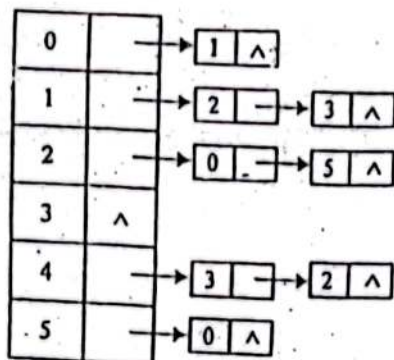
(2) 兄弟

6.

(1)

(2分)

(4分)

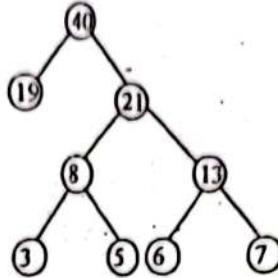


(2) 012534

(2分)

7. (1)

(4分)

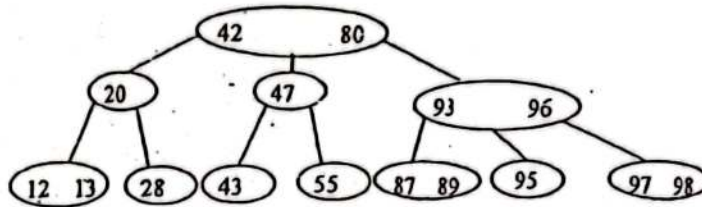


(2)  $WPL=(3+5+6+7)*3+19*1=82$

(2分)

8. (1). 插入 96

(第一次分裂 4分, 第二次分裂 2分)



四、程序填空题 (8分)

(每空 2分)

- (1)  $A[i] < A[\text{left}]$
- (2)  $A[j] > A[\text{left}]$
- (3)  $A[\text{left}], A[j]$
- (4)  $j+1, \text{right}$

五、程序阅读题 (6分)

- (1) 计算叶子结点数
- (2) 4

(3分)

(3分)

六、程序设计题 (12分)

(1)

```
template <class T> class SingleList;
```

```
template <class T>
```

```
class Node
```

```
{
```

(2分)

```

class SingleList
{ public:
    SingleList();
    ~SingleList();
    bool Insert(const T& x);
private:
    Node<T>* first;
    int length;
};

```

(2分)

```

(2) template<class T>
bool SingleList<T>::Insert(const T&x)
{

```

```

    Node<T>* p,q,r;
    p=first->link;
    q=first;
    while(!p->data==x)
    { q=p;
      p=p->link;
    }

```

(3分)

```

    if(p->data==x) return false;
    r=new Node<T>;
    r->data=x;
    r->link=q->link;
    q->link=r;
    length++;
    return true;
}

```

(2分)

(2分)

(1分)