

石家庄铁道学院 2009-2010 学年第 1 学期

**2008 级本科班期末考试试卷（A）答案**

课程名称： 数据结构 B      任课教师： 武守秋、邸书灵      考试时间： 120 分钟

学号： \_\_\_\_\_      姓名： \_\_\_\_\_      班级： \_\_\_\_\_

考试性质（学生填写）： 正常考试（ ） 缓考（ ） 补考（ ） 重修（ ） 提前修读（ ）

题 号	一	二	三	四	五	六	七	总分
满 分	20	20	45	15				100
得 分								
阅卷人								

**一、单项选择题（每小题 2 分，共 20 分）**

- 对于只在表的首、尾进行插入操作的线性表，宜采用的存储结构为：（ C ）。  
（A）顺序表                      （B）用头指针表示的单循环链表  
（C）用尾指针表示的单循环链表      （D）单链表
- 在  $n$  个结点的顺序表中，算法的时间复杂度是  $O(1)$  的操作是：（ A ）。  
（A）访问第  $i$  个结点（ $1 \leq i \leq n$ ）和求第  $i$  个结点的直接前驱（ $2 \leq i \leq n$ ）  
（B）在第  $i$  个结点后插入一个新结点（ $1 \leq i \leq n$ ）  
（C）删除第  $i$  个结点（ $1 \leq i \leq n$ ）  
（D）将  $n$  个结点从小到大排序
- 线性表若采用链式存储结构时，要求内存中可用存储单元的地址（ D ）。  
（A）必须是连续的                      （B）部分地址必须是连续的  
（C）一定是不连续的                      （D）连续或不连续都可以
- 线性表  $L$  在（ B ）情况下适用于使用链式结构实现。  
（A）需经常修改  $L$  中的结点值                      （B）需不断对  $L$  进行删除插入  
（C） $L$  中含有大量的结点                      （D） $L$  中结点结构复杂
- 有一个含头结点的双向循环链表，头指针为  $head$ ，则其为空的条件是：（ C ）。  
（A） $head \rightarrow prior == NULL$                       （B） $head \rightarrow next == NULL$   
（C） $head \rightarrow next == head$                       （D） $head \rightarrow next \rightarrow prior == NULL$
- 以下关于广义表的叙述中，正确的是：（ A ）。  
（A）广义表是由 0 个或多个单元素或子表构成的有限序列  
（B）广义表至少有一个元素是子表  
（C）广义表不能递归定义                      （D）广义表不能为空表
- 具有  $n(n > 0)$  个结点的完全二叉树的深度为（ C ）。  
（A） $\lceil \log_2(n) \rceil$       （B） $\lfloor \log_2(n) \rfloor$       （C） $\lfloor \log_2(n) \rfloor + 1$       （D） $\lceil \log_2(n+1) \rceil$

8. 某二叉树结点的中序序列为 A、B、C、D、E、F、G，后序序列为 B、D、C、A、F、G、E，则其左子树中结点数目为：（ C ）。
- （A）3 （B）2 （C）4 （D）5
9. 设一棵完全二叉树具有 1000 个结点，则此完全二叉树有（ A ）个叶子结点。
- （A）500 （B）499 （C）501 （D）498
10. 在一个无向图中，所有顶点的度数之和等于所有边数的（ C ）倍。
- （A）1/2 （B）1 （C）2 （D）4

## 二、填空题（每空 2 分，共 20 分）

1. 数据结构被形式地定义为  $(D, R)$ ，其中 D 是 数据元素 的有限集合，R 是 D 上的 关系 有限集合。
2. 数据项 是数据的不可分割的最小单位。
3. 向一个长度为 n 的顺序表的第 i 个元素( $1 \leq i \leq n+1$ )之前插入一个元素时，需向后移动  $n-i+1$  个元素。
4. 向栈中压入元素的操作是先 移动栈顶指针，后 存入元素。
5. 不包含任何字符（长度为 0）的串 称为空串；由一个或多个空格（仅由空格符）组成的串 称为空白串。
6. 假设有二维数组  $A_{6 \times 8}$ ，行列下标从 0，0 开始。每个元素用相邻的 6 个字节存储，存储器按字节编址。已知 A 的起始存储位置（基地址）为 1000，末尾元素  $A_{57}$  的第一个字节地址为 1282；若按行存储时，元素  $A_{14}$  的第一个字节地址为  $(8+4) \times 6 + 1000 = 1072$ 。

## 三、简答题（每小题 5 分，共 40 分）

1. 计算下列程序中  $x=x+1$  的语句频度

```
for(i=1;i<=n;i++)
    for(j=1;j<=i;j++)
        for(k=1;k<=j;k++)
            x=x+1;
```

【解答】 $x=x+1$  的语句频度为：

$$T(n) = 1 + (1+2) + (1+2+3) + \dots + (1+2+\dots+n) = n(n+1)(n+2)/6$$

2. 下面是出栈的算法，请把填上语句

//---- 栈的顺序存储表示 ----

```
#define STACK_INIT_SIZE 100; //初始分配量
#define STACKINCREMENT 10; //分配增量
typedef struct {
    SElemType *base; //栈底指针
    SElemType *top; //栈顶指针
    int stacksize; //栈容量
} SqStack;
```

```
Status Pop (SqStack &S, SElemType &e) {
    // 若栈不空，则删除 S 的栈顶元素，
```

```

// 用 e 返回其值，并返回 OK;
// 否则返回 ERROR
if (S.top == S.base) return ERROR;
e = *--S.top;
return OK;}

```

3. 写出下列程序段的输出结果（队列中的元素类型 QElem Type 为 char）。

```

void main() {
Queue Q; Init Queue (Q);
Char x='e'; y='c';
EnQueue (Q,'h'); EnQueue (Q,'r'); EnQueue (Q, y);
DeQueue (Q,x); EnQueue (Q,x);
DeQueue (Q,x); EnQueue (Q,'a');
while(!QueueEmpty(Q)){ DeQueue (Q,y);printf(y); }
Printf(x);
}

```

【解答】输出为“char”。

4. 已知 L 是无表头结点的单链表，且 P 结点既不是首元结点，也不是尾元结点，请写出在 P 结点后插入 S 结点的核心语句序列。

【解答】此题答案不唯一，但若从已给定序列中挑选，则限制颇多。

```

Q=P;
P=L;
while(P->next!=Q)P=P->next;
P=Q;
S->next=P->next;
P->next=S;

```

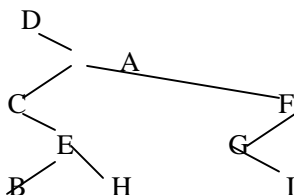
已知 P 结点，则不必“顺藤摸瓜”，直接链接即可。

S->next=P->next;

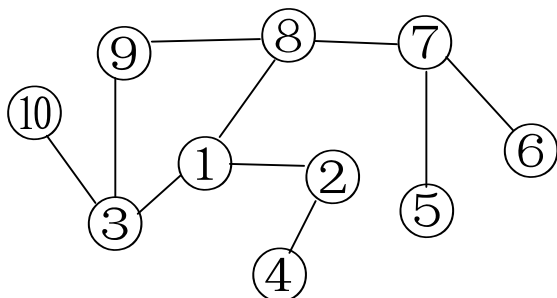
P->next=S;

5. 给定二叉树的两种遍历序列，分别是：前序遍历序列：D, A, C, E, B, H, F, G, I；中序遍历序列：D, C, B, E, H, A, G, I, F，试画出二叉树 B，并简述由任意二叉树 B 的前序遍历序列和中序遍历序列求二叉树 B 的思想方法。

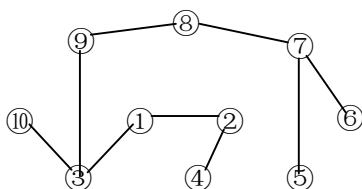
【解答】方法是：由前序先确定 root，由中序可确定 root 的左、右子树。然后由其左子树的元素集合和右子树的集合对应前序遍历序列中的元素集合，可继续确定 root 的左右孩子。将他们分别作为新的 root，不断递归，则所有元素都将被唯一确定，问题得解。



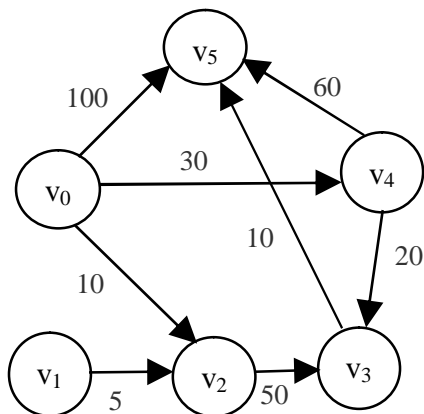
6. 下图是一个连通图，请画出以顶点①为根的深度优先生成树。（5分）



【解答】以顶点①为根的深度优先生成树：



7. 用迪杰斯特拉算法求源点  $v_0$  到其余各顶点的最短路径，请写出求解过程（7分）



【解答】

终点	从 $v_0$ 到各终点的 D 值和最短路径的求解过程				
	i=1	i=2	i=3	i=4	i=5
$v_1$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
					无

V2	10 (v0,v2)				
V3	$\infty$	60 (v0,v2,v3)	50 (v0,v4,v3)		
V4	30 (v0,v4)	30 (v0,v4)			
V5	100 (v0,v5)	100 (v0,v5)	90 (v0,v4,v5)	60 (v0,v4,v3,v5)	
Vj	V2	v4	v3	V5	
S	{v0,v2}	{v0,v2,v4}	{v0,v2,v3,v4}	{v0,v2,v3,v4,v5}	

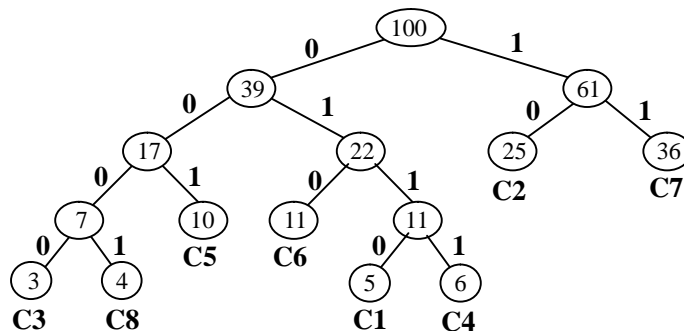
8. 假定用于通信的电文仅由 8 个字母 c1, c2, c3, c4, c5, c6, c7, c8 组成, 各字母在电文中出现的频率分别为 5, 25, 3, 6, 10, 11, 36, 4。试为这 8 个字母设计不等长 Huffman 编码, 并给出该电文的总码数。(8 分)

【解答】

已知字母集 { c1, c2, c3, c4, c5, c6, c7, c8 }, 频率 { 5, 25, 3, 6, 10, 11, 36, 4 }, 则 Huffman 编码为

c1	c2	c3	c4	c5	c6	c7	c8
0110	10	0000	0111	001	010	11	0001

电文总码数为  $4 * 3 + 4 * 4 + 3 * 10 + 3 * 11 + 4 * 5 + 4 * 6 + 2 * 25 + 2 * 36 = 257$



#### 四、算法设计 (15 分)

1. 写一个算法把输入的十进制数转换成二进制数。(7 分)

【解答】

```
void conversion( ) {
    InitStack(S); //构造空栈
    scanf ("%d", N);
```

```

        while(N){
            Push(S, N%2);
            N=N/2;
        }
    while(! StackEmpty(S))
        { Pop(S,e); printf("%d",e); }
} //conversion

```

2. 写一个中序遍历二叉树 T 的非递归算法，采用二叉链表作为存储结构。（8 分）

【解答】

```

typedef struct BiTNode { //结点结构
    TElemType    data;
    struct BiTNode  *lchild, *rchild;
    // 左右孩子指针
} BiTNode, *BiTree;
Status InOrderTraverse(BiTree T, Status (*Visit)(TElemType e))
{
    InitStack(S);
    Push(S, T); //根指针进栈
    While(!StackEmpty(S)){
        While(GetTop(S, p) && p) Push(S, p->lchild); //向左走到尽头
        Pop(S, p); //空指针退栈
        if(!StackEmpty(S)){ //访问结点,向右一步
            Pop(S, p); if(!Visit(p->data)) return ERROR;
            Push(S, p->rchild);
        } //if
    } //while
    Return OK; } //InOrderTraverse

```