



# 厦门大学《数据结构》期末试题·答案

考试日期：2011.1 (B)

信息学院自律督导部



一、(本题 10 分) 请根据下面的描述写出销售部门的数据结构(用 C 语言): 假设一个销售部门有  $n$  个职员(最多不超过  $N$  个,  $N$  为 100), 其中有一个是销售经理。每个职员都各自有一些客户, 客户的个数不固定, 不同职员的客户不重叠。

答案:

```
#define N 100

typedef struct CustNode {
    Customer    cust;
    struct CustNode * next;
} CustNode, * CustLink;

typedef struct {
    Saleman    sm;
    CustLink   firstcust; //指向第一个销售员
} SalemanNode;

typedef struct SaleDept {
    SalemanNode saleman[N];
    int    n; //销售员的人数
    int manager; //销售经理在数组 saleman[N]中的序号
} SaleDept;
```

二、(本题 15 分)

(1) 线性表和广义表的主要区别点是什么? 已知广义表:  $C=(a,(b,(a,b)), ((a,b),(a,b)))$ , 则  $\text{tail}(\text{head}(\text{tail}(C)))=?$

(2) 满足什么条件可以实施二分查找? 二分查找的时间复杂度是多少?

答:

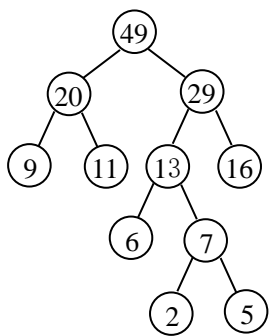
(1) 线性表和广义表都是元素  $a_1, a_2, \dots, a_n$  组成的序列, 其主要区别点在于: 在线性表中,  $a_i$  是单个元素(原子); 在广义表中,  $a_i$  可以是单个元素(原子), 也可以是广义表。

$\text{tail}(\text{head}(\text{tail}(C))) = ((a,b))$

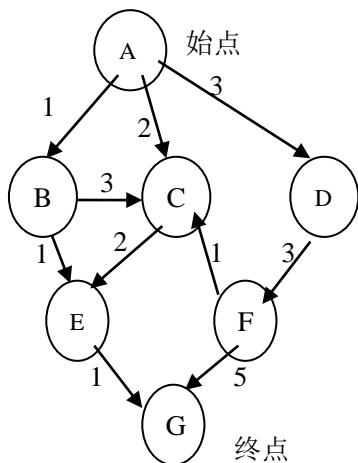
(2) 序列  $a_1, a_2, \dots, a_n$  必须在数组(顺序表)中, 且有序; 时间复杂度为  $O(\log n)$ 。

三、(本题 10 分) 给定一组权值 (5, 9, 11, 2, 6, 16), 试设计相应的哈夫曼树。

解: 构造的哈夫曼树如下图所示:

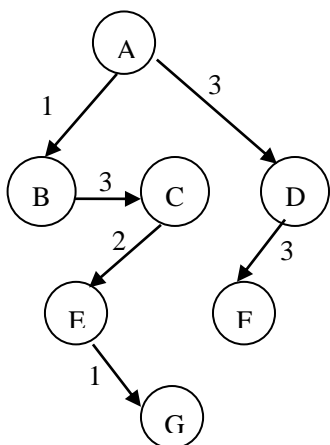


四、(本题 15 分) 某带权有向图如下:



- (1) 写出深度优先搜索结点访问序列, 并画出深度优先生成树 (当有多种选择时, 编号小的结点优先);
- (2) 写出该图的拓扑序列 (当有多种选择时, 编号小的结点优先);
- (3) 将该图作为 AOE 网络, 写出求关键路径的过程。

解: (1) 深度优先搜索顺序是: A, B, C, E, G, D, F  
深度优先生成树如下图所示。



(2) 该图的拓扑序列为: A, B, D, F, C, E, G

(3) 求解过程如下:

$$\begin{aligned}
 ve(A) &= 0 & ve(D) &= 3 & ve(F) &= ee(D) + 3 = 6 & ve(B) &= 1; \\
 ve(C) &= \max\{ ve(A) + 2, ve(B) + 3, ve(F) + 1 \} = 7 \\
 ve(E) &= \max\{ ve(B) + 1, ve(C) + 2 \} = 9
 \end{aligned}$$

$ve(G)=\max\{ ve(E)+1, ve(F)+5 \}=11$   
 $vl(G)=11$                        $vl(E)=vl(G)-1=10$                        $vl(C)=vl(E)-2=8$   
 $vl(B)=\min\{ vl(E)-1, vl(C)-3 \}=5$   
 $vl(F)=\min\{ vl(G)-5, vl(C)-1 \}=6$                        $vl(D)=vl(F)-3=3$   
 $vl(A)=\min\{ vl(B)-1, vl(C)-2, vl(D)-3 \}=0$

所以

$e(AB)=ve(A)=0$     $e(AC)=ve(A)=0$     $e(AD)=ve(A)=0$   
 $e(BC)=ve(B)=1$     $e(BE)=ve(B)=1$     $e(CE)=ve(C)=7$     $e(DF)=ve(D)=3$   
 $e(EG)=ve(E)=9$     $e(FC)=ve(F)=6$     $e(FG)=ve(F)=6$   
 $l(AB)=vl(B)-1=4$     $l(AC)=vl(C)-2=6$     $l(AD)=vl(D)-3=0$   
 $l(BC)=vl(C)-3=5$     $l(BE)=vl(E)-1=9$     $l(CE)=vl(E)-2=8$     $l(DF)=vl(F)-3=3$   
 $l(EG)=vl(G)-1=10$     $l(FC)=vl(C)-1=7$     $l(FG)=vl(G)-5=6$

所以

$e(AD)=l(AD)$     $e(DF)=l(DF)$     $e(FG)=l(FG)$

所以关键路径为：ADFG

五、（本题 10 分）已知待散列存储的关键字序列为（4,15,38,49,33,60,27,71），哈希函数为  $H(key)=key \text{ MOD } 11$ ，哈希表 HT 的长度为 11，采用线性探测再散列法解决冲突。试构造此哈希表，并求出在等概率情况下查找成功的平均查找长度。

解：哈希表为

0	1	2	3	4	5	6	7	8	9	10
33				4	15	38	49	60	27	71

平均查找长度为： $(1+2+2+3+1+4+5+6)/8=3$ 。

六、（本题 15 分）在两个有序线性表中，寻找是否存在共同元素。如果存在共同元素，返回第一个共同元素在第一个有序表中的位置。请设计数据结构，并在其上设计算法。

答：可以参考有序表的归并算法。

数据结构可以使用一维数组，并且第 0 个元素放空。

int SearchCommonItem(int a[n], int b[m])//第 0 位放空，返回值为 0 代表找不到

```

{
    int i=1,j=1;
    while (i<=n && j<=m)
    {
        if (a[i]==b[j]) return i;
        else if (a[i]<b[j]) i++;
        else j++;
    }
    return 0;
}

```

七、（本题 10 分）编写一个遍历二叉查找树 T 的算法，要求遍历过程恰好按结点键值从大到小的次序进行。二叉树 T 数据结构采用二叉链表。

答案：

```

void travel(bitree bt)
{
    if(bt){
        travel(bt.rchild); visit(bt.data); travel(bt.lchild);
    }
}

```

```

}
}

```

八、(本题 15 分) 一个正整数序列存放在带头结点的链表 L 中，每个结点存放一个正整数。请编写算法将该链表调整为所有奇数在链表的前部分，所有偶数在链表的后部分，并且调整后的奇数序列和偶数序列都与它们在原来序列中的次序一致(例如：原序列 1 2 3 4 5 6，调整后 1 3 5 2 4 6)。要求：除算法外，还要给出数据结构、算法思想和代码注释。

[参考答案]

```

typedef struct Node {
    int data;
    struct Node * next;
}*List;

```

[方法 1]

链表 r 保存的是倒序的奇数序列，然后逆序插入链表 L 的头部。

```

void OddEven( List L)
{ struct Node *p, *q, *r;
  p = L; r = NULL; //r 是不带头结点的链表
  while( p->next!=NULL )
    if( p->next->data%2==0 ) p = p->next;//如果是偶数，不管
    else {
        q = p->next;          //把结点从链表 L 中移走
        p->next = q->next;    //把结点从链表 L 中移走
        q->next = r;          //加入到链表 r 中
        r = q;                //加入到链表 r 中
    }
  while( r!=NULL ) {          //链表 r 逆序插入链表 L 的头部
    q = r;r = r->next;
    q->next = L->next;L->next = q;
  }
}

```

[方法 2]

顺序扫描链表直到表尾，每找到一个奇数时将其从原来位置删除，然后插入到奇数部分的表尾辅助指针 ot (初始指向虚设的头结点) 之后，并更新 ot 指针令其指向该新插入结点

```

void OddEven( List L)
{ struct Node *p, *q, *ot;
  p = L; ot = L;
  while( p->next!=NULL )
    if( p->next->data%2==0 ) p = p->next;//如果是偶数，不管
    else if (p!=ot)
    { q = p->next;          //把结点从链表 L 中移走
      p->next = q->next;    //把结点从链表 L 中移走
      q->next = ot->next;   //到链表 ot 之后
      ot->next=q;           //到链表 ot 之后
      ot = q;
    }
}

```