厦门大学《数据结构》期末试题·答案

考试日期：2010.1（cyj）　　信息学院自律督导部

一、（本题 10 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 tail(head(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 $a_1,a_2,\ldots,a_n$ 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。

tail(head(tail(C))) = ((a,b))

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

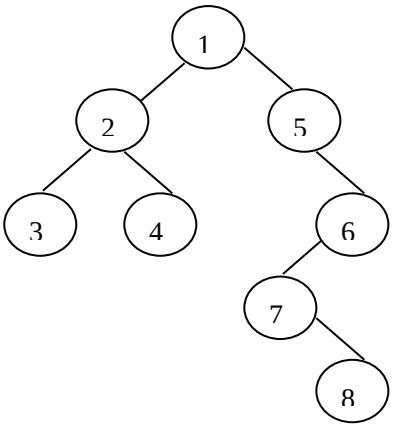
二、（本题 10 分）假定用于通讯的电文仅由 a、b、c、d、e、f、g 等 8 个字母组成，字母在电文中出现的频率分别为：0.07、0.19、0.02、0.06、0.32、0.03、0.21 和 0.10。试为这些字母设计哈夫曼编码。

答：一种编码如下：

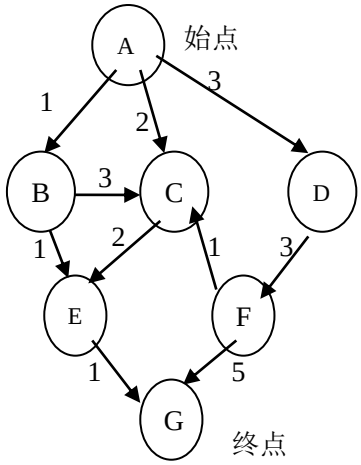
a: 0010　b: 10　c:00000　d:0001　e:01　f:00001　g:11　h:0011

三、（本题 10 分）一棵二叉树的先序、中序和后序序列分别如下，部分未显示，请画出该二叉树。先序序列: _ 2 3_5_7 8;中序序列: 3_4 1_7 8 6；后序序列：_ 4 2 _ _ 6 5 1。

答：



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。）

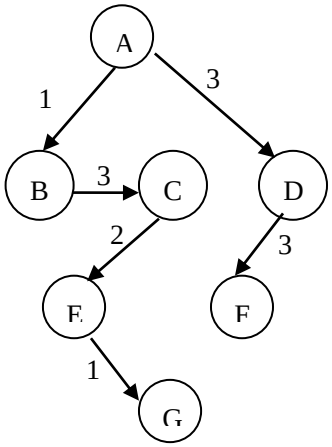
（2）写出该图的拓扑序列（当有多种选择时，编号小的结点优先。）

（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：

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

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

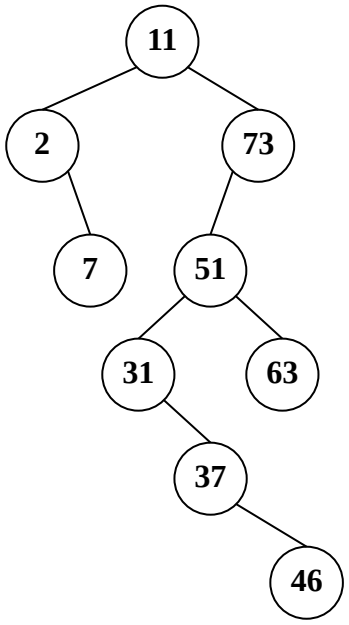
所以

e(AD)=l(AD)　e(DF)=l(DF)　e(FG)= l(FG)

所以关键路径为：ADFG

五、（本题 15 分）设有一个关键字序列{11,73,51,31,63,37,46,2,7}，（1）从空树开始构造排序二叉树，画出得到的排序二叉树；（2）计算该排序二叉树在等概率下查找成功的平均查找长度。（3）如果要获取这些关键字的**从大到小**的排列，要对该排序二叉树使用何种遍历方法，如何处理？

答：（1）



（2）对该排序二叉树进行：先遍历右子树，再遍历根，后遍历左子树。

六、（本题 15 分）（15 分）设关键字序列为：49，38，66，80，70，15，22，欲对该序列进行从小到大排序。

（1）用直接插入排序法进行排序，写出每趟的结果。

（2）采用待排序列的第一个关键字作为枢轴，写出用快速排序法的一趟和二趟排序之后的状态。

（3）假设有个系统要多次对 n 个关键字进行排序，n 很大且每次排序时关键字的分布情况不明。系统不希望每次排序时间变动过大，而且希望越快越好，哪种排序算法较好？为什么？

答：

（1）直接插入排序法

原始关键字序列为：(49)　38　66　80　70　15　22
(38　49) 66　80　70　15　22
(38　49　66) 80　70　15　22
(38　49　66　80) 70　15　22
(38　49　66　70　80) 15　22
(38　49　66　70　80) 15　22
(15　38　49　66　70　80) 22
(15　22　38　49　66　70　80)

（2）快速排序法
原始关键字序列为：49，38，66，80，70，15，22
第一趟排序后　　22　38　15　(49)　70　　80　　66
第二趟排序后　　15　(22)　38　　66　　(70)　80
（3）堆排序算法较好。首先，因为堆排序平均时间复杂度为 O(nlogn)，性能较好；另外，因为关键字分布情况不明，堆排序最坏时间复杂度为 O(nlogn)，保证计算时间不会出现 $O(n^2)$而造成很大的延迟，此时，快速排序算法不太合适。此外，堆排序只需要 O(1)个辅助空间，归并排序虽然总体性能也很不错，也满足系统的要求，但需要 O(n)的辅助空间。

七、（本题 15 分）设 L 是一个带头结点的非递减有序单链表的表头指针，试设计一个算法，将元素 e 插入到链表 L 中的合适地方，使得该链表仍是非递减有序。
答：

```
typedef struct Node {
    ElemType e;
    struct Node * next;
} Node, *LinkList;
void Insert( LinkList * L, ElemType e)
{ Node * p, *q, *s;
    p = L->next; q = L;
    while ( p!=NULL && p->data <= e )
        q = p; p = p->next;
    }
    s = (Node *) malloc( sizeof(Node) );
    s->data = e;
    s->next = p;
    q->next = s;
}
```

八、（本题 10 分）请利用两个队列 Q1 和 Q2 来模拟一个栈。已知队列的三个运算定义如下：bool EnQueue(Queue &Q,int e):插入一个元素 e 入队列；bool DeQueue(Queue &Q,int &e):删除一个元素 e 出队列；bool QueueEmpty(Queue Q)：判队列为空。假设数据结构 Queue 已定义，栈 Stack 的数据结构定义如下。请利用队列的运算来实现该栈的三个运算：Push(Stack ST,int x):元素 x 入 ST 栈；Pop(Stack ST, int x)：ST 栈顶元素出栈，赋给变量 x；StackEmpty(Stack　ST)：判 ST 栈是否为空。

```
typedef struct {
    Queue Q1;
    Queue Q2;
} Stack;
```

答：队列 Q1 或 Q2 中的某一个保存所有的栈元素。
程序如下：

```
bool StackEmpty(Stack S)
{
```

```
	return QueueEmpty(S.Q1) && QueueEmpty(S.Q2);
}

bool Push(Stack &S, int e)
{
	if( QueueEmpty(S.Q2)==false )
		return EnQueue(S.Q2, e);
	return EnQueue(S.Q1, e);
}

bool Pop( Stack &S, int &e)
{Queue *from,*to;
 int x;
	if( QueueEmpty(S.Q1)==true && QueueEmpty(S.Q2)==true ) return false;
	if( QueueEmpty(S.Q1)==false ) {
		from = &S.Q1; to = &S.Q2;
	}
	else {
		from = &S.Q2; to = &S.Q1;
	}

	DeQueue(*from,x);
	while( QueueEmpty(*from)==false ) {
		EnQueue(*to,x);
		DeQueue(*from,x);
	}
	e = x;
	return true;
}
```