厦门大学《数据结构》期末试题·答案

考试日期：2012.1（A）　　信息学院自律督导部

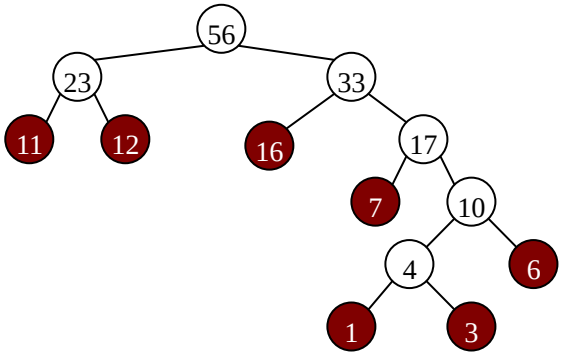
一（10 分）
1)线性表的两种存储结构各有什么优缺点？
2) 利用 GetHead 和 GetTail 操作，从广义表((((apple),pear),banana),orange)中得出 banana。

解：（第一小题 6 分，第二小题 4 分）
1) 顺序存储是按索引（如数组下标）来存取数据元素，优点是可以实现快速的随机存取，缺点是插入与删除操作将引起元素移动，降低效率。对于链式存储，元素存储采取动态分配，利用率高。缺点是须增设指针域，存储数据元素不如顺序存储方便。优点是插入与删除操作简单，只须修改指针域。
2)设 L=((((apple),pear),banana),orange)
所求操作为：GetHead【GetTail【GetHead【L】】】

二、（10 分）栈与队列的区别和共同点是什么？图的深度优先探索和广度优先搜索分别适用上述哪种结构，并简单说明理由？

解：（第一部分 6 分，第二部分 4 分。）
1)区别：栈在表尾进行插入和删除，具有“后进先出”的特点；队列在表尾插入，在表头删除，具有“先进先出”的特点；
共同点：栈和队列都是特殊的线性表，都是 n 个数据元素的有限序列，都是数据结构的逻辑结构。
2)图的深度优先探索适用栈，节点需要后进先出；图的广度优先探索适用队列，节点需要先进先出。

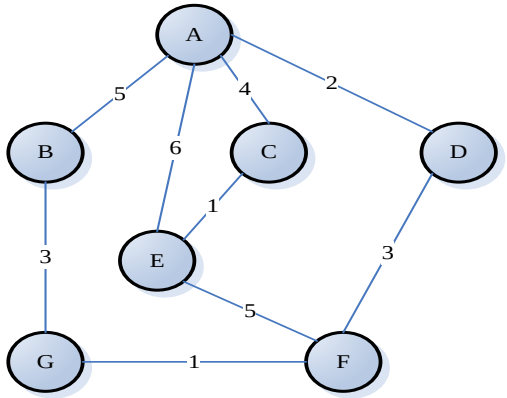
三、（10 分）给定权值集合{1, 3, 6, 7, 11, 12, 16}，构造相应的哈夫曼树并计算带权路径长度。
解：构造的哈夫曼树如下（8 分）



其带权路径长度=16*2+12*2+11*2+7*3+6*4+3*5+1*5=143（2 分）

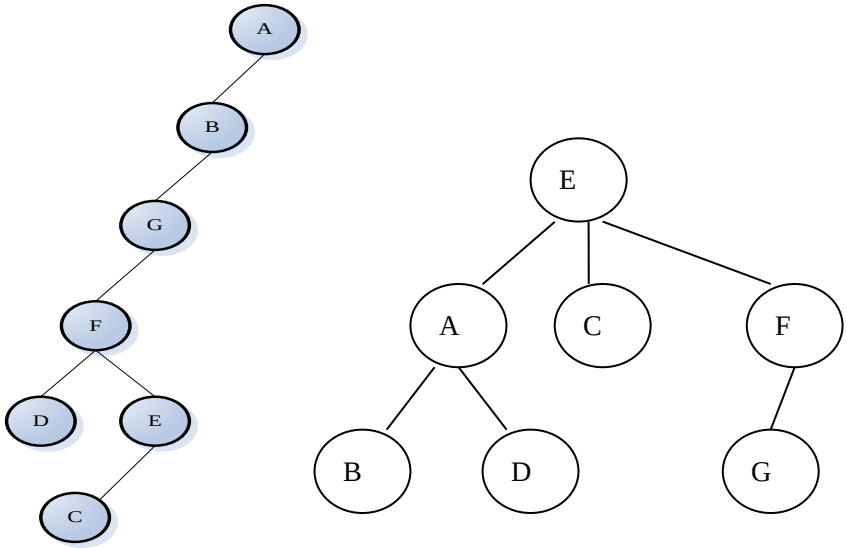
四、（15 分）考虑下图：

1) 从顶点 A 出发，求它的深度优先生成树(按照字符顺序进行访问)。
2) 从顶点 E 出发，求它的广度优先生成树(按照字符顺序进行访问)。
3) 使用普里姆算法，求它的最小生成树（给出树的生成过程）。

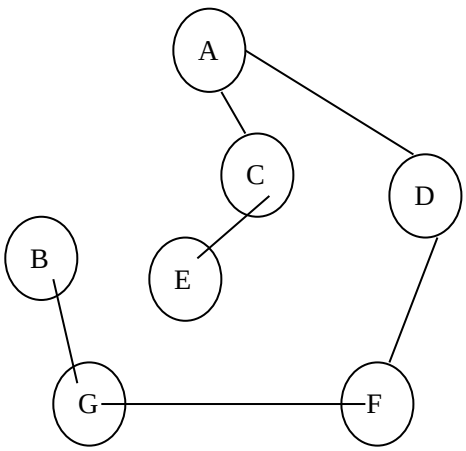


解：（每小题 5 分）

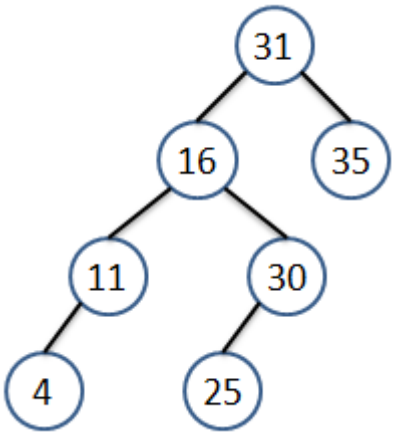
深度优先生成树和广度优先生成树分别为左右两图:



最小生成树最终结果为：

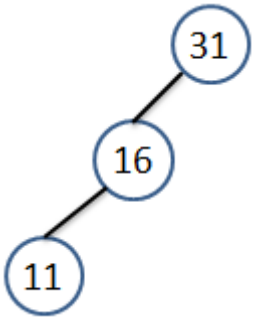


五、（15 分）设关键字序列为 31,16,11,35,30,25,4，回答下列问题：
1)请画出依次插入该序列后的二叉排序树。
2)在等概率的情况下，该二叉排序树查找成功的平均查找长度。
3)请画出依次插入该序列后的平衡二叉排序树。
解：（第一小题 5 分，第二小题 3 分，第三小题 7 分）
1)插入上述序列之后的二叉排序树：

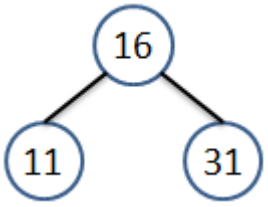


2)查找成功的平均查找长度为：(1+2*2+3*2+4*2)/7=19/7=2.7

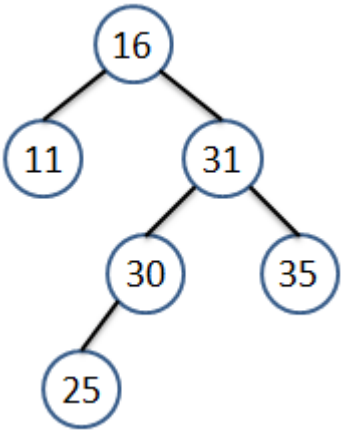
3) 插入 31,16,11 之后：



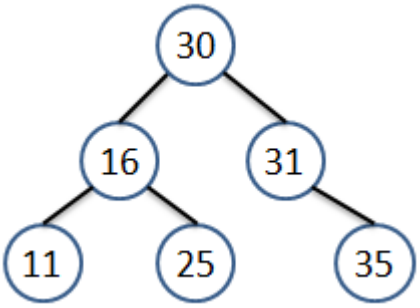
调平衡后得到 AVL：



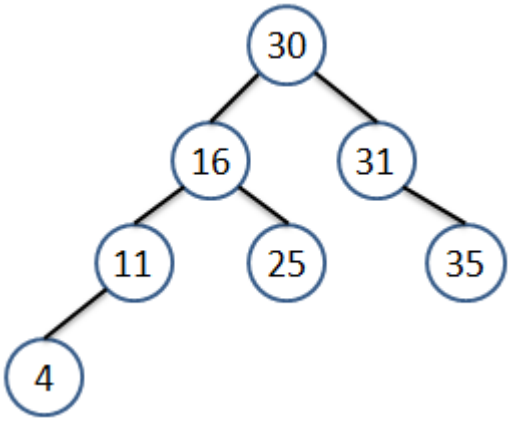
再插入 35,30,25 之后：



调平衡后得到 AVL：



再插入 4 之后得到 AVL：



六、（10 分）设哈希表的地址范围为[0,10]，哈希函数 H(key)=(key^2+2) MOD 11，现在要将数据 4,7,3,6,8,9,2 依次插入到哈希表中。

1)使用线性探测再散列法处理冲突，请画出相应的哈希表，并计算查找成功的平均查找长度。

2)使用链地址法处理冲突，请画出相应的哈希表，并计算查找成功的平均查找长度。

【参考答案】（第一小题 6 分，第二小题 4 分）

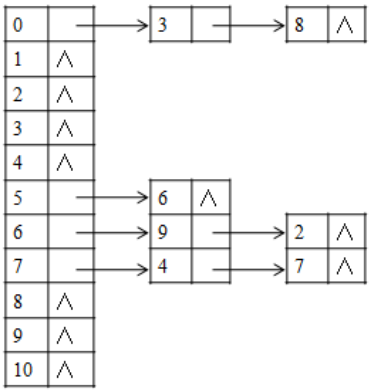
（1） 使用线性探测再散列法来处理冲突所构造的哈希表：

地址	0	1	2	3	4	5	6	7	8	9	10
数据	3	8				6	9	4	7	2	

H(4)=7，H(7)=7，H(3)=0，H(6)=5，H(8)=0，H(9)=6，H(2)=6

查找成功 ASL = (1+2+1+1+2+1+4)/7=12/7=1.71

（2） 使用链地址法来处理冲突所构造的哈希表：



查找成功 ASL = (1*4+2*3)/7 = 10/7 = 1.43

七、（12 分）对于关键字序列（28，25，36，5，17，30，2，51，10，25，46，59）进行从小到大排序，写出下列排序算法第一趟的执行结果： 1)起泡排序；2)初始增量为 5 的希尔排序；3)快速排序；4)堆排序（给出堆顶为最大值的大堆即可）。

答：每小题各 3 分。

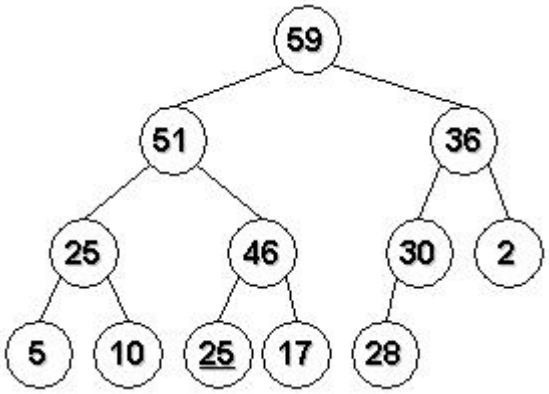
手工执行下列排序算法，第一趟执行结束时得到的关键字序列如下：

1）起泡排序：25，28，5，17，30，2，36，10，25，46，51，59

2） 初始增量为 5 的希尔排序：28，2，36，5，17，30，25，51，10，25，46，59

3） 以第一个元素为枢轴的快速排序：25，25，10，5，17，2，28，51，30，36，46，59

4）堆排序（给出堆顶为最大值的初始堆）：59，51，36，25，46，30，2，5，10，25，17，28



八、（8 分）有一个单链表，其结点的元素值以递增顺序排列，给出数据结构，并编写一个算法删除该单链表中元素值相同的结点。

参考答案：
从头到尾扫描单链表，若当前结点与后继结点的值不相同，则指针后移，若相同则删除该后继结点。

```
typedef struct LNode{
    ElemType data;
    struct LNode *next;
}LNode, *LinkList;
//带头结点

Status Del_Dup(LinkList &L)
{
    p=L->next;   //p 指向第一个元素
    if (p ==NULL) return; //  处理空表
    while (p->next!=NULL) {
        if(p->data!=p ->next->data)         //若当前结点与后继结点的值不相同，则指针后移
            p=p->next;
        else{   //若当前结点与后继结点的值相同，则删除该后继结点
            q=p->next;
            p->next=q->next;
            free(q);
        }
    }
}
```

或者使用两个指针：

```
Status Del_Dup(LinkList &L)
{
```

```
    if (L->next ==NULL) return; //  处理空表
    f=L->next; p=f->next; //在扫描过程中，f 保持为 p 的前驱
    while (p) {
        if(f->data!=p ->data){          //若当前结点与后继结点的值不相同，则指针后移
            f=p;
            p=p->next;
        }
        else{   //若当前结点与后继结点的值相同，则删除该后继结点
            q=p;
            f->next=p->next;
            free(q);
            p=p->next;
        }
    }
}
```

九、（10 分）在 n 个元素中，找出第 k 大的元素，给出数据结构，并设计算法实现上述要求，并给出时间复杂性分析，最好是在 O(n)的时间复杂性之内。

答：可以借鉴快排算法来达到 O(n) 。

```
static ITEM_select(ITEM [] a, int l, int r, int k)
{
     while(r>l)   {
          int i = partition(a, l ,r); //partition 后，比 a[i]小的都在 i 左边，
                                      //比 a[i]大的都在  i 右边。
          if(i==k) return a[k];
          if(i<k) r=i-1; //从 l 到 i-1 做 selection
          if(i>k) l=i+1; //从 i+1 到 r 做 selection
     }
}
```

对于足够大的随机数组，每次 partion 会把数组分成大约相等的两半，那么每次问题 size 缩小一般，比较次数为 n+n/2+n/ 4+.....+1=2n。因此为 O(n)。

注意 select 和 quicksort 不同，因为 quicksort 每次都要比较 N，而 select 的比较次数是指数减少的，因此是 O(n)而不是 O(nlogn)