厦门大学《数据结构》期末试题·答案

考试日期：2009•1（A）　　信息学院自律督导部

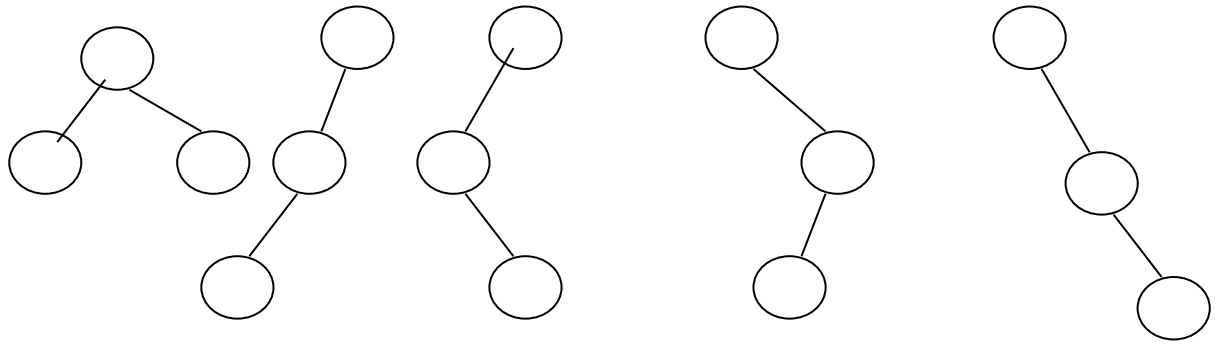
一、（本题 10 分）栈与队列的区别和共同点是什么？队列主要有哪两种物理实现？
答：

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。

队列的物理实现：链队列和循环队列。

二、（本题 10 分）给出二叉树的定义，并画出具有 3 个结点的二叉树的所有形态。
答：所谓二叉树或者是空树，或者是如下定义的树：每个结点至多只有两棵子树，并且二叉树的子树有左右之分，其次序不能颠倒。
3 个结点的二叉树的所有形态如下：

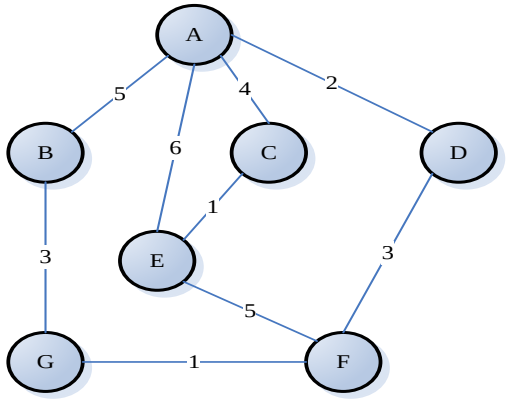


三、(本题 15 分)考虑下图：

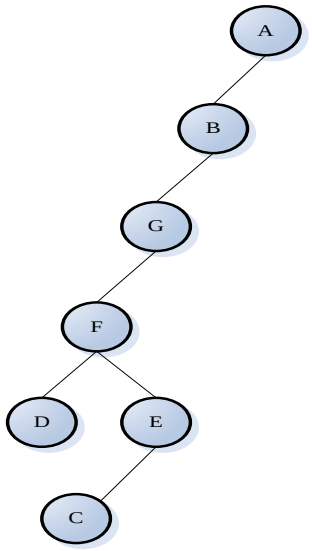
1）从顶点 A 出发，求它的深度优先生成树(字母小的优先访问)。

2）从顶点 E 出发，求它的广度优先生成树(字母小的优先访问)。

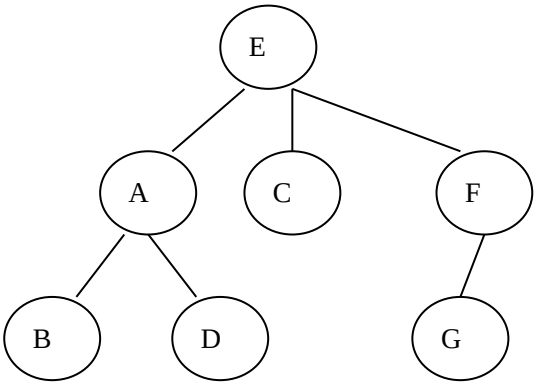
3）使用克鲁斯卡尔算法，求它的最小生成树（给出树的生成过程）。



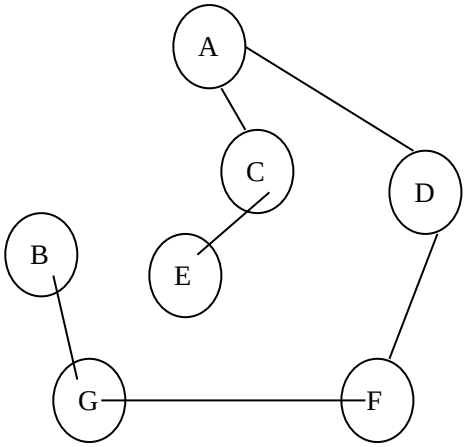
答：深度优先生成树为:



广度优先生成树为:



最小生成树为:（因为有相同的权值，树的生成过程可能不相同）



评分标准：每小题 5 分。

四、（本题 15 分）假定一个待哈希存储的线性表为(32,75,29,63,48,94,25,46,18,70)，哈希地址空间为 0～12，若采用除留余数法 H(K)=K % 13 构造哈希函数，并使用链地址法处理冲突，试画出最后得到的哈希表，并求出平均查找长度。

解：

元素	32	75	29	63	48	94	25	46	18	70
哈希地址	6	10	3	11	9	3	12	7	5	5
查找次数	1	1	1	1	1	2	1	1	1	2

平均查找长度为（1*8+2*2）/10=1.2
评分标准： 画出哈希表得 10 分，平均查找长度得 5 分。

五、（本题 15 分）已知键值序列为 {45,56,83,31,72,35,14,47,89,19}，要求给出：

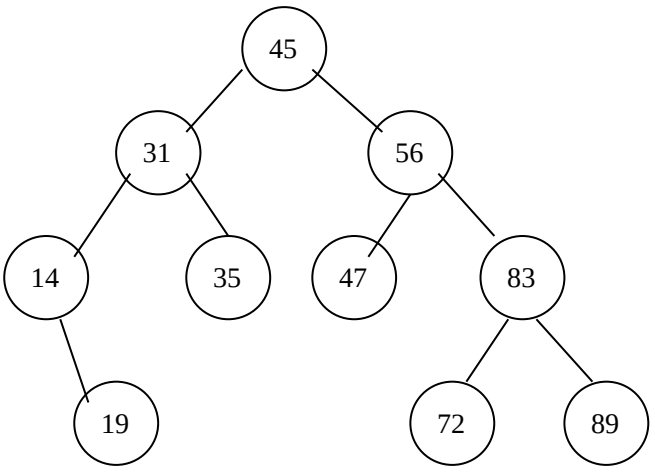
（1） 按键值排列次序构造一棵二叉排序树。

（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） 针对上述 10 个键值，在不同的排列次序下所构造出的不同形态的二叉排序树中，在最坏和最好情况下，二叉排序树的高度各是多少？

解：总分为 15 分，每一小步 5 分。

（1）

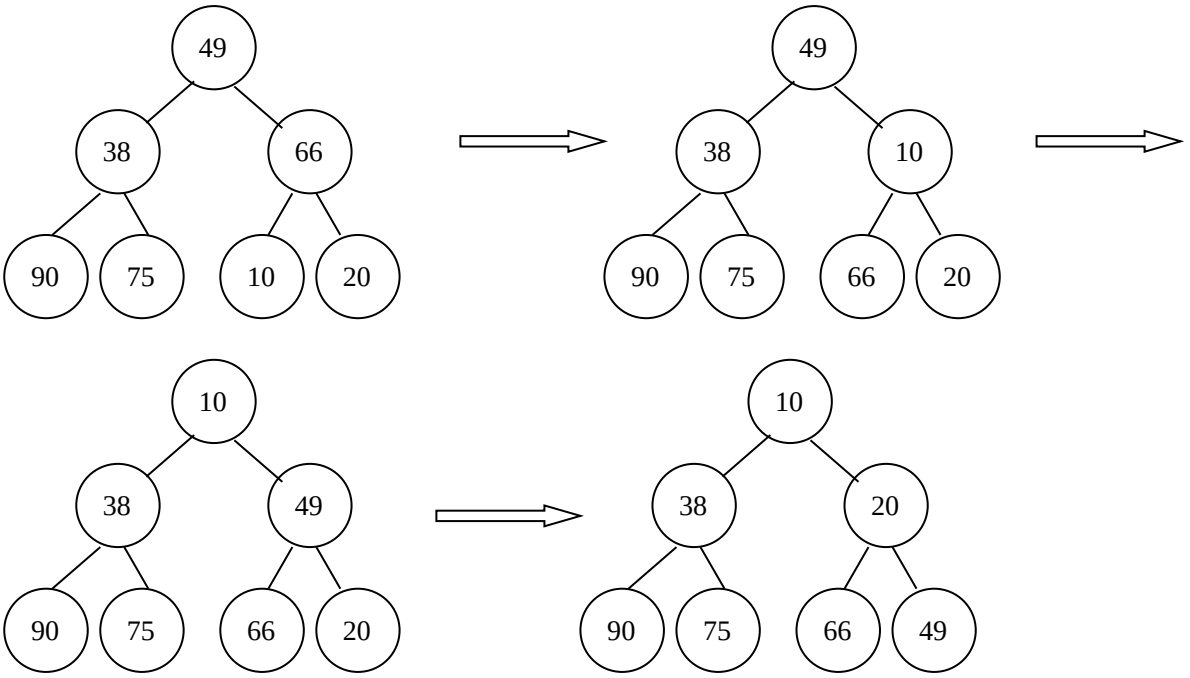


（2）在等概率情况下，该二叉排序树的平均检索长度是：

ASL=(1+2*2+3*4+4*3)/10=29/10=2.9

（3）对于上述 10 个键值，在最坏情况下，每个结点（除了叶子结点）只有右孩子（或者只有左孩子），高度为 10。在最好情况下，高度为$\lfloor \log_2 10 \rfloor +1=4$。

六、（本题 10 分）设关键字序列为：49，38，66，90，75，10，20。把这些关键字调整成堆顶元素取最小值的堆（写出过程）。



七、（本题 10 分）试设计一个递归算法（函数），判断二叉树 T 是否是满二叉树，假设 T 是以二叉链表存储。

```
typedef struct BiTNode{
    TElemType data;
    Struct BiTNode *lchild, *rchild;
} BiTNode, *BiTree;
```

解答：

算法：

（1）如果二叉树 T 是空树，则是满二叉树；

（2）如果二叉树 T 非空，左子树或右子树不是满二叉树，则不是满二叉树；

（3）如果二叉树 T 非空，左右子树都是满二叉树，但深度不一样，则 T 不是满二叉树。

（4）如果二叉树 T 非空，左右子树都是满二叉树，而且深度一样，则 T 是满二叉树；

```
//该函数判断二叉树 T 是否是满二叉树
//如果是满二叉树，返回 TRUE，Depth 返回该树的深度；
//否则返回 FALSE，Depth 无定义；
Boolean Check( BiTree T, int &Depth)
{ int ldepth, rdepth;
    if( T==NULL) { Depth=0; return TRUE; }
    if( Check(T->lchild, ldepth)==FALSE ) return FALSE;
    if( Check(T->rchild,rdepth)==FALSE) return FALSE;
    if( ldepth!=rdepth ) return FALSE;
    Depth=ldepth+1; return TRUE;
```

}

八、（本题 15 分）在 n 个元素中，找出第 k 大的元素，最好是在 O(n)的时间复杂性之内。请设计数据结构，并在其上设计算法（函数），并给出时间复杂性分析。

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)