

哈尔滨工业大学

2000 年硕士研究生入学考试试题参考答案

考试科目：数据结构(覆盖高级语言)

报考专业：计算机科学与技术

考试科目代码：[419]

主观问答题，可根据考生表述明确与否酌情给分。

题号	一	二	三	四	五	六	七	八	九	十	十一	总分
分数	12	12	10	6	6	6	8	8	8	10	14	100

一、名词解释（每小题 3 分，共 12 分）

1. 抽象数据类型：一个数学模型以及定义在该模型上的操作集合的总称。
2. 算法的时间复杂性：算法中基本操作重复执行的次数是问题规模 n 的某个函数 $f(n)$ ，算法的时间量度记做 $T(n)=O(f(n))$ 。它表示随问题规模 n 的增大，算法执行的时间的增长率与 $f(n)$ 的增长率相同，称作算法的渐进时间复杂性，简称时间复杂性。
3. 散列法(hashing)：采用一种称为散列表的数据结构表示的数据集合，用这种方法进行查找，不比较关键字，而借助于一个函数(称为散列函数)，把要查找的元素关键字转换成一个数值，称为散列值或散列地址，按照散列地址确定要查找的元素在散列表中的位置。
4. 索引文件：除文件数据区本身以外，另建立一张指示逻辑记录和物理记录之间一一对应关系的索引表，这类包括文件数据区和索引表两大部分的文件称作索引文件。

二、填空题（每空 1 分，共 12 分）

1. 为了处理空表的方便
2. $n-1$
3. 某种遍历序列的直接前驱结点 某种遍历的直接后继结点
4. 数组表示 邻接表表示 左右链表示
5. 简单的排序算法（气泡，插入，选择） 快速排序 归并排序 堆排序 基数排序

三、判断题（每题 2 分，共 10 分）

1. (☒) 比如仅一个根结点的树。
2. (☒) 二叉树结点的度可以为 0、1、2。
3. (☒) 因为连通图上各边的权值均不相同，所以按 Prim 算法，每次选取最小权值的边，顺序必然是一定的，最小生成树必唯一。

4. (X) 无向图的邻接矩阵一定是对称矩阵，有向图的邻接矩阵可为对称矩阵，也可
为非对称矩阵。

5. (✓)

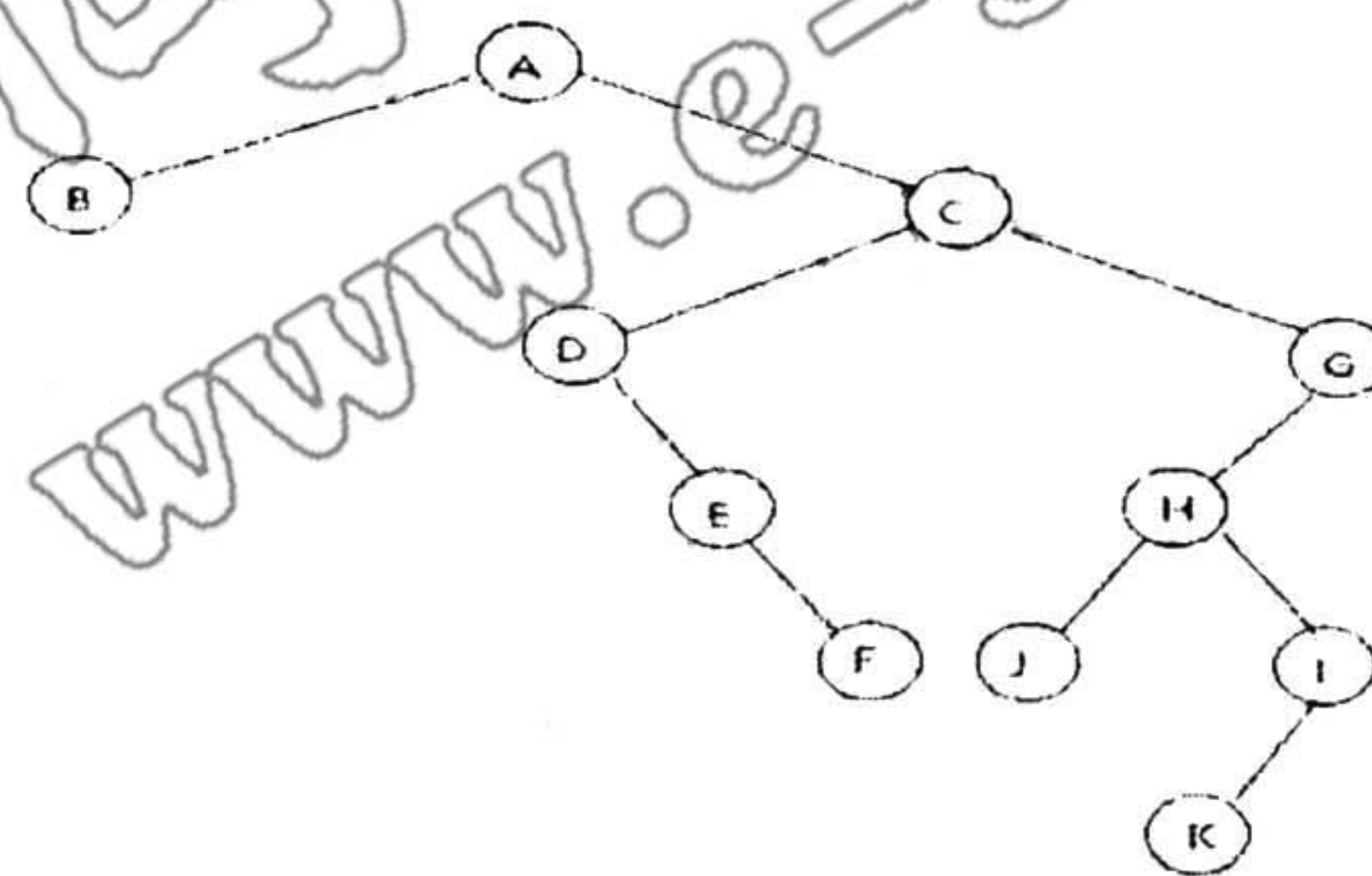
四、堆与二元查找树的区别：

堆与二元查找树均满足任一结点元素值小于其左右儿子的值，但是若按中根顺序遍历一棵二元查找树，即可得到递增顺序序列，而堆无此性质，需经过整理才能得到最终结果。

五、快速分类法的基本思想：

快速排列是由起泡排列改进而得的，它的基本思想是：在待排序的 n 个记录中任取一个记录（通常取第一个记录），将该记录放入最终位置后，数据序列被此记录分割成两部分。所有关键字比该记录关键字小的放置在前一部分，所有比它大的放置在后一部分，并把该记录排在这两部分的中间，这个过程称作一次快速排序。之后对所有的两部分分别重复上述过程，直至每一部分内只有一个记录为止。简而言之，每趟使表的第一个元素入终位，将表一分为二，对子表按递归方式继续这种划分，直至划分的子表长为 1。

六、其森林如图所示：



七、依次读入数据元素序列{a, b, c, d, e, f, g}进栈每进一个元素，机器可要求下一个元素进栈或弹栈，如此进行则栈空时弹出的元素构成的序列是以下那些序列？（8分）

第一个和第四个正确。

对于第一个：{d, e, c, f, b, g, a}

Push : a b c d e f g

Pop : d e c f b g a

对于第四个：{c, d, b, e, f, a, g}

Push : a b c d e f g

Pop : c d e f a g

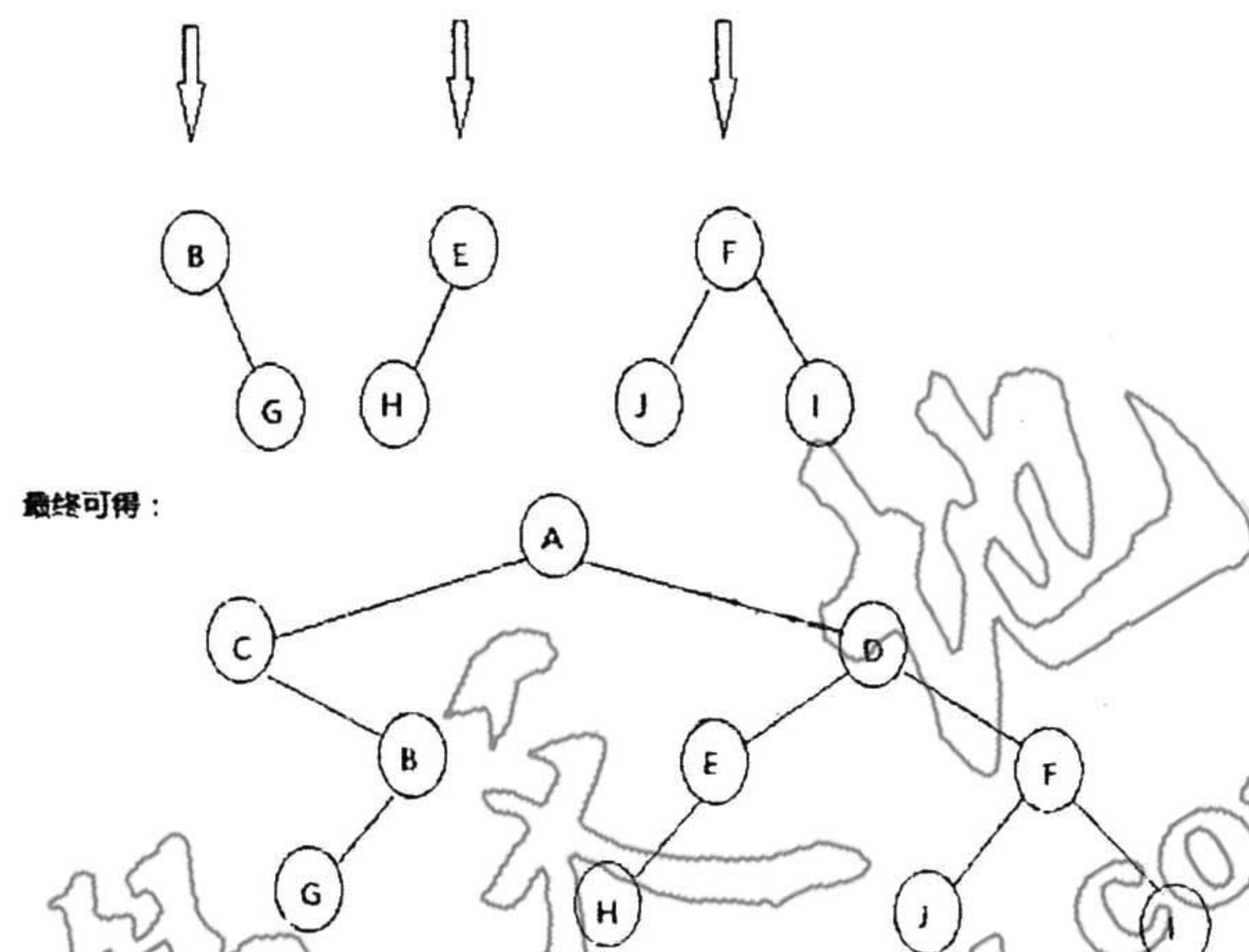
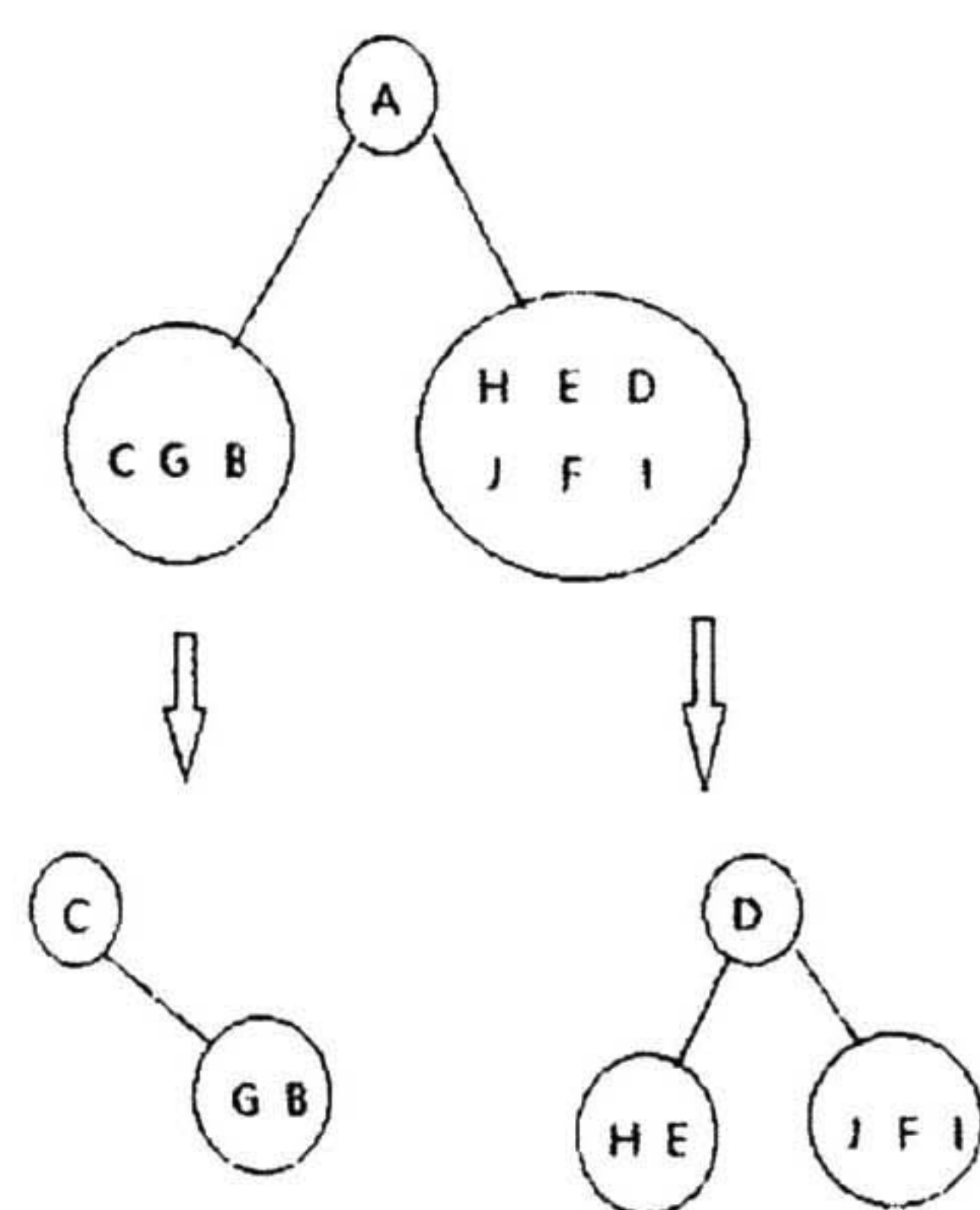
八、

若已知先根和后根的遍历结果，不能构造出这颗树。因为叶子无兄弟的时候，有二义性，因此知道先根和中根遍历或者知道后跟和中根遍历的结果才能构造出这颗二元树。

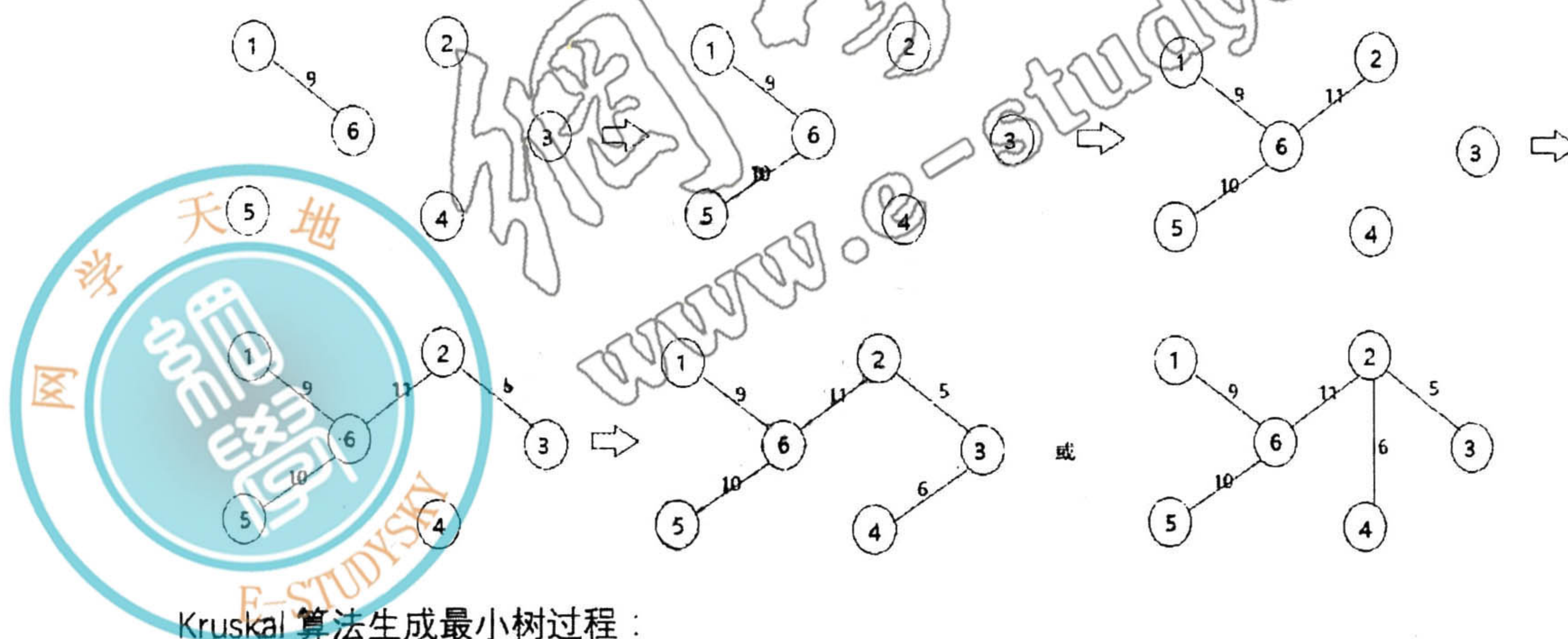
中根：CGBA HEDJFI

后根：GBCHEJIFDA

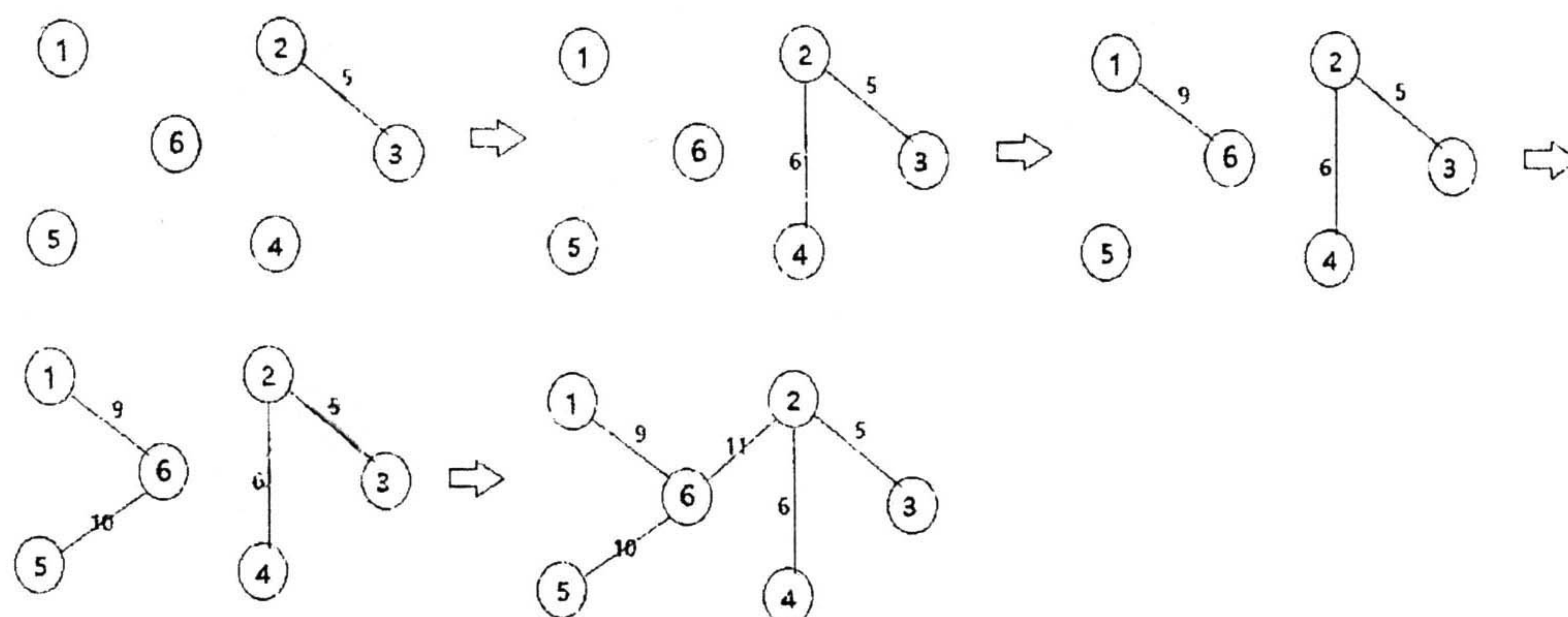
根据后根遍历，最后一个元素为树的根，根据中根遍历根左侧的为其左子树，右侧的为其右子树，对左右子树分别递归上述过程，即可得到原树。



九、Prim 算法生成最小树过程



Kruskal 算法生成最小树过程：



十、(10 分)

结点的型：

```
struct celltype
{
```



```
elementtype data;
celltype *LCHILD, *RCHILD;
};
树型：
typedef celltype *BTREE;
设计思想：1.中根遍历 2.利用栈倒序排列输出
void SEARCH(BTREE BT)
```

```
{
    stack S, R; BTREE T;
    elementtype p;
    MAKENULL(S);
    MAKENULL(R);
    while (!ISEMPTY(T) || EMPTY(S))
    {
        if (!ISEMPTY(T))
        {
            push(T, S);
            T = LCHILD(T);
        }
        else
        {
            T = TOP(S);
            POP(S);
            push(DATA(T), R);
            T = RCHILD(T);
        }
    }
    while (!EMPTY(R))
    {
        p = TOP(R);
        POP(R);
        printf(R)
    }
}
```

十一、(14分)

结点的型：

```
struct node {
    datatype data;
    node *lchild;
    node *rchild;
};
```

树型：

```
typedef node *BinTree; //二元树结点的结构
```


哈工大计算机考研全套视频和资料，真题、考点、典型题、命题规律独家视频讲解！

详见：网学天地（www.e-studysky.com）；咨询QQ：2696670126

算法要点：在完全二叉树中没有这样的结点，它有右子树但没有左子树；也没有这样的结点，它没有孩子，但它的右边兄弟和它下一层的结点有孩子。由此特征来判断二叉树是否为完全二叉树。

算法描述：

```
int JudgeCompleteTree(BinTree bt){
    //判断bt二叉树是否是完全二叉树，如果是返回1，如果不是返回0.
    int tag = 0;
    BinTree p = bt;
    Queue q;
    if (p == NULL) return 0;
    Enqueue(q, p);
    while (!QueueEmpty(q)){
        p = Dequeue(q);
        if (p->lchild && !tag)
            Enqueue(q, p->lchild);
        else if (p->rchild)
            return 0;
        else tag = 1;
        if (p->rchild && !tag)
            Enqueue(q, p->rchild);
        else if (p->lchild)
            return 0;
        else tag = 1;
    }
    return 1;
}
```