5/9

判断下列说法是否正确：高为5 （不含叶子层）的3阶B-树最少有31个关键字

一棵m阶B树是一棵平衡的m路搜索树。它或者是空树，或者是满足下列性质的树： 1、根结点至少有两个子女； 2、每个非根节点所包含的关键字个数 j 满足：┌m/2┐ - 1 <= j <= m - 1； 3、除根结点以外的所有结点（不包括叶子结点）的度数正好是关键字总数加1，故内部子树个数 k 满足：┌m/2┐ <= k <= m ； 4、所有的叶子结点都位于同一层。 由m阶B-树性质可知，根结点至少有两棵子树，根结点之外的所有非终端结点至少有m/2棵子树：则三阶B-树的形状至少类似于一棵满二叉树，也即高度为5的三阶B-树至少有(25-1=)31个结点。 因此题中描述是正确的

B树不属于二叉树

哈夫曼树的节点要么是叶子节点，要么是度为2的节点，不可能出现度为1的节点

000，001，010，011，1

B

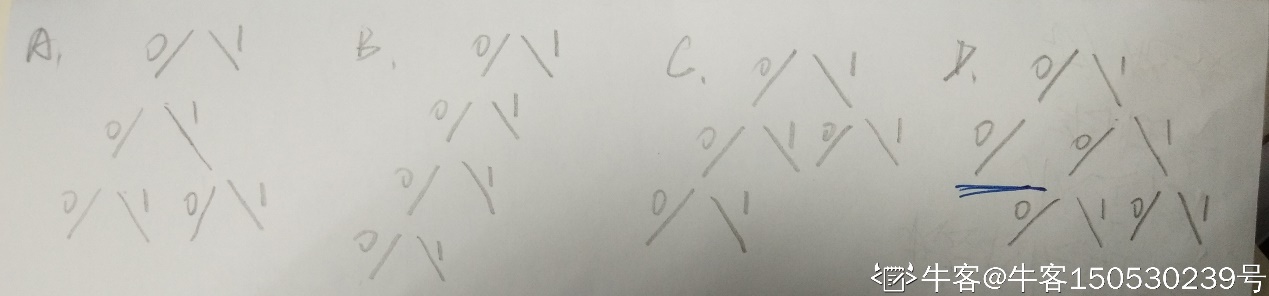
0000，0001，001，01，1

C

000，001，01，10，11

D

00，100，101，110，111



在一棵度为3的树中，度为3的节点个数为2，度为2的节点个数为1，则度为0的节点个数为6

设该树总共有n个节点,则n=n0+n1+n2+n3. 该树中除了根节点没有前驱以外,每个节点有且只有一个前驱,因此有n个节点的树的总边数为n-1条.根据度的定义,总边数与度之间的关系为：n-1=0\*n0+1\*n1+2\*n2+3\*n3. 联立两个方程求解,可以得到n0=6

有以下5个叶子节点1,1,3,2,5构成的哈夫曼树的带权路径长度为25

结点带权路径长度=该结点到根的路径长度×该结点权

哈夫曼树:给定叶子权值和叶子数 可以构造出不同结构的二叉树 其中带权路径长度最小的二叉树称为最优二叉树

哈夫曼树是一种最优二叉树

哈夫曼树算法

根据n个权值构造具有n棵二叉树的森林---森林中的每颗二叉树都只有一个根结点 结点的数据域为一个权值 该结点左右子树都是空

在森林中选出2棵根结点权值最小的树A B(这样的树不止2棵---任选2棵)---创建一个新结点作为A B的根结点 A

B分别作为根结点的左右孩子(权值小的在左边大的在右边 这样规定 哈夫曼树就唯一了)---将A

B的权值相加作为根结点的权值---森林中产生一颗新树C(:~有左右孩子各1)

在森林中继续选出2棵根结点权值最小的树(2.中产生的C也在选择范围内)---重复2.的过程

直到将森林中所有的数合并成一颗二叉树---这颗二叉树就是哈夫曼树

按照上述算法可以得到如下哈夫曼树 其带权路径长度= 5\*1+3\*2+2\*3+(1+1)\*4 =25 选D

A picture containing text, handwriting, sketch, drawing

Description automatically generated

设一组权值集合 W=(15, 3, 14, 2, 6, 9, 16, 17)，要求根据这些权值集合构造一棵哈夫曼树，则这棵哈夫曼树的带权路径长度为

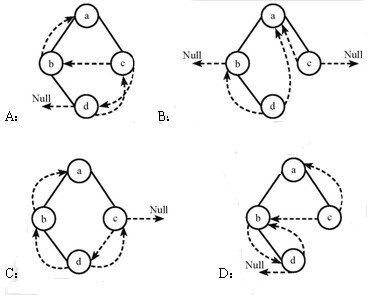
A picture containing text, screenshot, diagram

Description automatically generated

对一棵二叉树进行后续遍历，其输出结果为A，B，C，这样的二叉树有\_5\_\_\_棵

二叉树的形态个数即为卡特兰数h(n)=C(2n,n)/(n+1) (n=0,1,2,...)

下列线索二叉树中（用虚线表示线索），符合后序线索树定义的是d



（1）先忽略虚线部分，只看实线，进行后序遍历，得dbca前驱后继关系

（2）考虑每个结点，当左孩子为空时，指向前驱；右孩子为空时指向后继。（用虚线）

结点d，无左右孩子，则指向前驱null，后继b

结点b, 无左孩子，则指向前驱d

结点c，无左右孩子，则指向前驱b，后继a

结点a,, 有左右孩子。

根据以下性质： 3阶B-树每个节点至多有三个子节点，除叶子节点外其他节点至少有两个子节点。 所有叶子节点都在同一层，叶结点可以有一个或两个关键码。 三个子节点的非叶子节点含有两个关键码，有两个子节点的非叶子节点有一个关键码。 结合题目中的“至多”关键字可以得出每层节点1,3,9,27,81，2+3\*2+9\*2+27\*2+81\*2=242个关键字

在线索化二叉树中，节点t没有左子树的充要条件是t->ltag==1

下列选项给出的是从根分别到达两个叶结点路径上的权值序列，能属于同一棵哈夫曼树的是

24,10,5 和 24,10,7

B

24,10,5 和 24,12,7

C

24,10,10 和 24,14,11

D

24,10,5 和 24,14,6

在哈夫曼树中，左右孩子权值之和为父结点权值。仅以分析选项 A 为例：若两个 10 分别属于两棵不同的子树，根的权值不等于其孩子的权值和，不符；若两个 10 属同棵子树，其权值不等于其两个孩子（叶结点）的权值和，不符。 B 、 C 选项的排除方法一样

5/10/2023

设森林F对应的二叉树为B，它有m个结点，B的根为p，p的右子树结点个数为n，森林F中第一棵树的结点个数是m – n

B树、B-树、B+树、B\*树相关

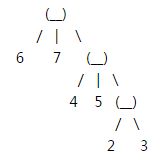
<https://www.cnblogs.com/hellowooorld/p/7344866.html>

对一棵二叉排序树进行\_\_\_\_\_\_\_\_\_\_\_\_遍历得到的结点序列是一个有序序列 中序

已知三叉树T中6个叶结点的权分别是 2, 3, 4, 5, 6, 7，T 的带权（外部）路径长度最小是（）46

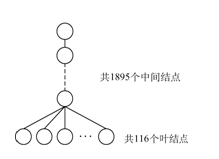
这道题的考点是哈夫曼树的一个性质：权值越大的结点距离根节点越近。

所以只要从上往下，按三叉树的样式先把数值大的结点排好，就可以了



已知一棵有 2011 个结点的树，其叶结点个数为 116 ，该树对应的二叉树中无右孩子的结点个数是1896

树转换为二叉树时，树中每一个分支结点的所有子结点中的最右子结点无右孩子，根结点转换后也没有右孩子，因此，对应的二叉树中无右孩子的结点个数 = 分支结点数 +1=2011 - 116+ 1=1896



将森林转换为对应的二叉树，若在二叉树结点中，结点m是结点n的双亲结点的双亲结点，则在原来的森林中，m和n可能具有的关系是\_\_\_\_。1和3

1.父子关系

2.m的双亲结点与n的双亲结点是兄弟关系

3.兄弟关系

普通树转换为二叉树的方法是：（http://blog.csdn.net/zm1\_1zm/article/details/77222460）

①树中所有相同双亲结点的兄弟节点之间加一条连线

②对树中不是双亲结点第一个孩子的结点，只保留新添加的该结点与左兄弟结点之间的连线，删去该结点与双亲结点之间的连线

③整理所有保留和添加的的连线，使每个结点的第一个孩子结点连线位于左孩子指针位置，使每个结点的右兄弟结点连线位于右孩子指针位置

Prim算法(适合稠密图，贪心算法的运用，时间复杂度O(n+e),邻接表存储；O(n^2),图 ) 2.Kruskal算法(适合稀疏图，贪心算法的运用，时间复杂度O(eloge),e为边数 ) 求图最短路径算法： 1.DFS/BFS(单源) 2.Floyed算法(多源) 3.Dijkstra算法(单源) 4.Bellman-Ford算法(单源，负权) 相信你能看懂！

把一棵树转换为二叉树后,这棵二叉树的形态是

唯一的

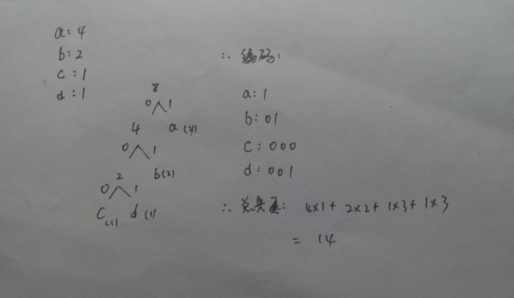
当向二叉排序树中插入一个结点，则该结点一定成为叶子结点。

从根开始按层序遍历一颗二叉树，需要使用（）作为辅助数据结构

仔细想一下，先将根结点A入队，再将A出队（遍历到A），然后将A的所有子结点（假设为B、C）入队，接着，B出队（遍历到B），将其所有子结点入队，接下来出队的自然是C（遍历到C），然后将C的所有子结点入队，这样，就遍历了两层。接下来出队的自然是B的第一个子结点（遍历到该节点），开始第三层的遍历……可见，我们需要一个先进先出的结构，自然选择队列

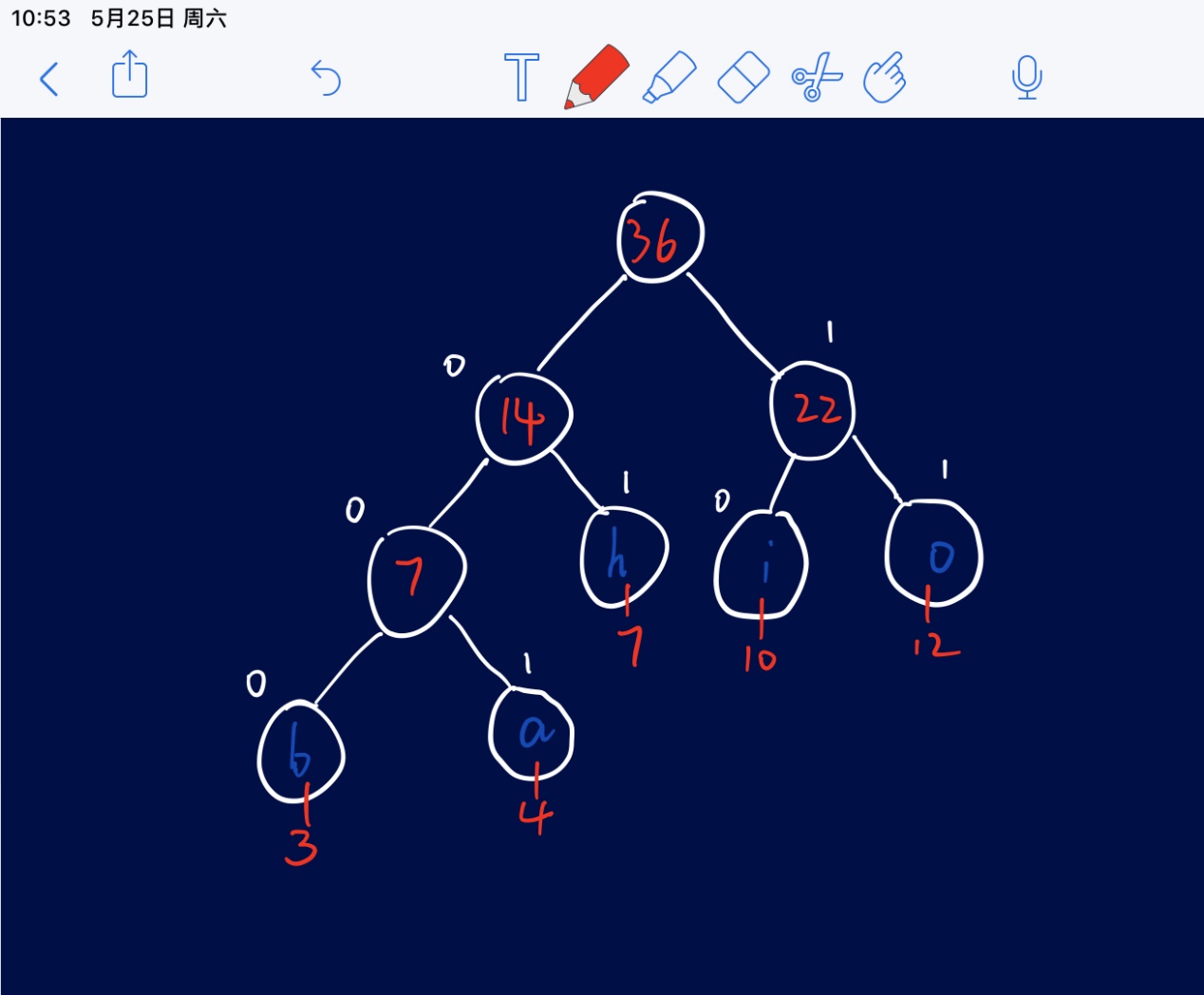
用二进制来编码字符串 "abcdabaa"，需要能够根据编码，解码回原来的字符串，最少需要()长的二进制字符串14

这道题需要对 abcd 进行 Huffman 编码。首先根据权值建立 Huffman 树，得到最优编码：a=0, b=10, c=110, d=111，然后数一下就行了。



度数为2的树：树的所有节点的度都小于等于2，且至少有一个节点的度为2。 二叉树：树的所有节点都必须小于等于2。（差别在于可以没有度数为2的节点）

某段文本中各个字母出现的频率分别是{a:4,b:3,o:12,h:7,i:10}，使用哈弗曼编码，则哪种是可能的编码



已知二叉树A(B(,D(F,H)),C(,E(G(I)))),由此二叉树转换的森林描述正确的是 该森林包含三棵树

以A为根的树有三个孩子

二叉树转换为森林

从根结点开始 若右孩子存在，则把与右孩子结点的连线删除，再查看分离后的二叉树，若右孩子存在，则连续删除除……，直到所有右孩子连线都删除为止，得到分离的二叉树

再将每棵分离后的二叉树转换为树

二叉树转换为树

二叉树转换为树是树转换为二叉树的逆过程

加线。若某结点的左孩子结点存在，则将这个左孩子的右孩子结点、右孩子的

右孩子结点、右孩子的右孩的右孩子结点...., 是左孩子的n个右孩子结点都作为此结点的孩子。将该结点与这些右孩子结点用线连接起来。

去线。删除原二叉树中所有结点与其右孩子结点的连线。

层次调整

给定n个带权结点，其Huffman树的结构不是唯一的

二叉平衡查找树：

左子树中所有节点的值小于根的值，右子树中的所有节点的值大于根的值

左右子树的高度之差的绝对值为0或1

森林变成二叉树：森林中每棵树都变成二叉树，然后再合起来。

而树变成二叉树：所有非叶节点，其第一个孩子会变为二叉树中的左孩子。

因此反过来说，二叉树中没有左孩子的节点，在原来的森林中一定是叶子节点。

若有一个叶子结点是二叉树中某个子树的中序遍历结果序列的最后一个结点，则它一定是该子树的前序遍历结果序列的最后一个结点。

对于一个共有n个结点、K条边的森林，共有几棵树 n – K

假设某棵二叉查找树的所有键均为1到10的整数，现在我们要查找5。下面\_\_\_\_不可能是键的检查序列 2,8,6,3,7,4,5

10,9,8,7,6,5

B

2,8,6,3,7,4,5

C

1,2,9,3,8,7,4,6,5

D

2,3,10,4,8,5

E

4,9,8,7,5

二叉排序树或者是一棵空树，或者是具有下列性质的二叉树： （1）若左子树不空，则左子树上所有结点的值均小于它的根节点点的值； （2）若右子树不空，则右子树上所有结点的值均大于或等于它的根结点的值； （3）左、右子树也分别为二叉排序树； （4）没有键值相等的节点。 先看A选项，第一个节点值为10，比要查找的5大，所以要接下来要比较10的左子树，左子树上的值一定比10小，而选项A中的第二个查找值是9，合理。此时，9仍然比待查找的数字5大，要继续查找9的左子树，左子树比9小，而查找的下一位数字为8，合理。依次类推。。。。

怎样遍历二叉查找树可以得到一个从小到大的有序序列 中序遍历

5/11/2023

在平衡二叉树中插入一个结点后造成了不平衡，设最低的不平衡结点为A，并已知A的左孩子的平衡因子为0右孩子的平衡因子为1，则应作( RL ) 型调整以使其平衡

<http://blog.csdn.net/zhanghaotian2011/article/details/8459281>

二叉树在线索化后，仍不能有效求解的问题是 后序线索二叉树中求后序后继

前序求后继，后序求前驱，中序啥都行

B+树不同于B树的特点之一是 能支持顺序查找

hash索引仅满足“=”、“IN”和“<=>”查询，不能使用范围查询

B

hash索引无法被用来进行数据的排序操作

C

对于组合索引，Hash索引在计算Hash值的时候是组合索引键合并后再一起计算Hash值，而不是单独计算Hash值，所以通过组合索引的前面一个或几个索引键进行查询的时候，Hash索引也无法被利用

D

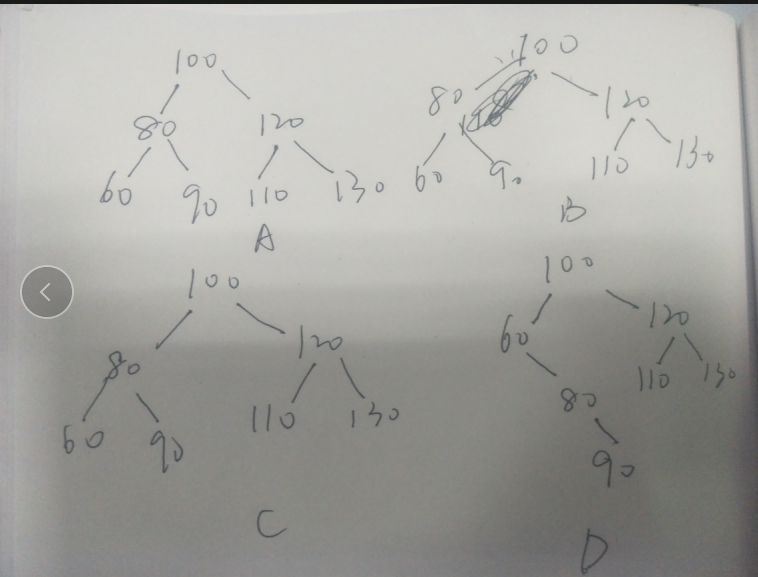
Hash 索引遇到大量Hash值相等的情况后性能并不一定就会比B-Tree索引高

平衡二叉树一定是二叉排序树

用三叉链表作二叉树的存储结构，当二叉树中有n个结点时，有()个空指针 n+2

特殊值法 n = 1, 双亲指针，2个孩子指针

分别以下列序列构造二叉排序树，与用其它三个序列所构造的结果不同的是



设F是由T1、T2和T3三棵树组成的森林，与F对应的二叉树为B，T1、T2和T3的结点数分别为N1、N2和N3，则二叉树B的根结点的左子树的结点数为（）N1-1

http://www.cnblogs.com/zhuyf87/archive/2012/11/04/2753950.html

若一个具有N个结点，M条边的无向图构成一个森林，(N>M)， 则该森林必有( )棵树

由树的性质得知：

树中边数等于结点数。

两个树合并成一个森林时，结点数不变，边数-1。

森林边树=森林结点树-树的个数，即树的个数为N-M

一颗完全二叉树第六层有8个叶结点（根为第一层），则结点个数最多有（）个

二叉树第k层最多有2的(k-1)次方个节点 第六层最多有32个节点 第五层最多有16个节点 第四层最多有8个节点 第三层最多有4个节点 第二层最多有2个节点 第一层最多有1个节点 完全二叉树的叶节点只可能出现在后两层 如果完全二叉树有6层，则前5层是满二叉树，总节点数目为16+8+4+2+1+8=39 如果完全二叉树有7层，则前6层是满二叉树， 前六层总节点数目为32+16+8+4+2+1=63 第六层有8个叶子节点，则有32-8=24个非叶子节点 第七层最多有24\*2个叶子节点 总节点数目为63+24\*2=111

将一棵树转换为二叉树后，这棵二叉树的形态是 一定没有右子树

如果T1是由有序树T转换而来的二叉树，那么T中结点的先序序列就是T1中结点的（）序列 先序

一棵3阶B-树中含有2047个关键字，包含叶结点层，该树的最大深度为12

http://haicang.blog.51cto.com/2590303/1134864

n个结点的线索二叉树上含有的线索数为

一个有n个节点的线索二叉树，每个节点都有指向左右孩子的两个指针域，则共有2n个指针域，而n个节点共有n-1条分支，所以共有2n-(n-1)个空指针域，即有n+1个线索

中序遍历二叉排序树得到的才是有序的序列

5/12/2023

考察的是哈夫曼树的特征和形成的过程。如给定4个叶子结点a，b，c和d，分别带权7，5，2和4，形成的哈夫曼树如下图所示： 所以A、B正确。初始森林中的n棵二叉树，每棵树有一个孤立的结点，它们既是根，又是叶子。 n个叶子的哈夫曼树要经过n-1次合并，产生n-1个新结点。最终求得的哈夫曼树有2n-1个结点。

5/13/2023

有一颗二叉树的前序遍历和后续遍历分别是1,2,3,4和4,3,2,1，则该二叉树的中序遍历可能是1,2,3,4 2,3,4,1 4,3,2,1

5/14/2023

在任意一棵非空二叉排序树中，删除某结点后又将其插入，则所得二排序叉树与原二排序叉树相同 只有被删除结点是叶子结点时命题才正确

完全二叉树：只有最下面的两层结点度能够小于2，并且最下面一层的结点都集中在该层最左边的若干位置的二叉树。

所以，叶子节点可以出现在最下面两层上

已知一棵完全二叉树的第6层（设根为第1层）有8个叶结点，则该完全二叉树的结点个数最多是

完全二叉树比满二叉树只是在最下面一层的右边缺少了部分叶结点，而最后一层之上是个满二叉树，并且只有最后两层有叶结点。第6层有叶结点则完全二叉树的高度可能为6或7，显然树高为7时结点更多。若第6层上有8个叶结点，则前六层为满二叉树，而第7层缺失了8×2=16个叶结点，故完全二叉树的结点个数最多为(27-1)-16=111个结点

已知中序遍历的序列为abcdef，高度最小的可能的二叉树的叶子是

ABC 高度最小的二叉树是平衡二叉树，也就是c为根节点或者d为根节点 当c为根节点时，左子树的叶子节点可能是a或者b中的一个，右子树的叶子节点是d和f，可能是adf或者bdf 当d为根节点时，左子树的叶子节点可能是a和c，右子树的叶子节点是e或f中的一个，可能是ace或者acf

5/16/2023

n个结点的线索二叉树上含有的线索数为 n＋l

图G的一棵最小代价生成树的代价未必小于G的其它任何一棵生成树的代价

因为最小生成树不唯一，所以可能相等

在一个空的5阶B-树中依次插入关键字序列{6,8,15,16,22,10,18,32,20}，插入完成后，关键字6所在结点包含的关键字个数为

如果B-树的阶数为m，则结点中关键字个数的范围为ceil(m/2)-1 ~ m-1个，所以关键字个数为2~4个。

过程如下

A screenshot of a computer

Description automatically generated with medium confidence

给定一棵二叉搜索树，根据节点值大小排序所需时间复杂度是线性的

某二叉树结点的中序序列为BDAECF，后序序列为DBEFCA，则该二叉树对应的森林包括（）棵树 3

二叉树转化为深林的前提，先判断根结点有没有右孩子，有就可以转化为深林，没有右孩子能转化为一颗树。

从根结点开始，如果有右子树，将右子树与根结点的连线删除，在查看分离后的二叉树，若有右孩子，则连线删除，递归这个过程，直到所有的右孩子连线都删除为止，得到分离的二叉树。分离二叉树转化为树，得到的树的个数即为森林包括的树的数量。

那么二叉树如何转化为树？

二叉树转化为树的过程：

1.加线。若某个结点的左孩子结点存在，将左孩子的右孩子结点作为此结点的孩子，即将该节点与右孩子结点用连线连接起来。

2.去线。删除原二叉树中所有结点与其右孩子结点的连线。得到的便是树

现有一棵无重复关键字的平衡二叉树（AVL 树），对其进行中序遍历可得到一个降序序列。下列关于该平衡二叉树的叙述中，正确的是

树中最大元素一定是无左子树