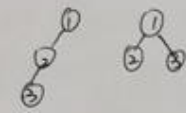
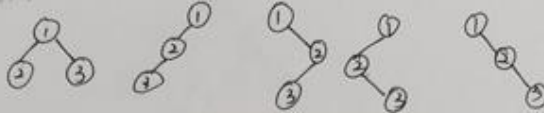


1、由3个结点可以构造出多少种不同的树（请画出）？由3个结点可以构造出多少种不同的二叉树（请画出）？



树：2种

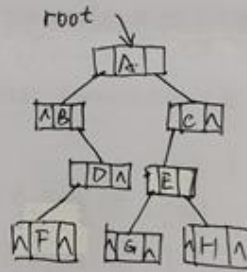
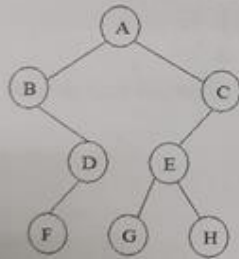


二叉树：5种

2、若对含 n 个结点的完全二叉树从上到下且从左至右进行 1 至 n 的编号，则：（1） n 号结点的双亲编号是多少？（2）最后一个非终端结点（非叶子结点）编号是多少？（3）度为 1 的结点具有什么特征？（4）最小的叶子结点编号是多少？

- (1) 双亲编号为 $\frac{n}{2}$ (向下取整)
- (2) 最后一个非终端结点为 $\frac{n}{2}$
- (3) 该结点处的子结点个数为 1
- (4) 最小的叶子结点编号为 $\frac{n}{2} + 1$

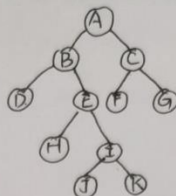
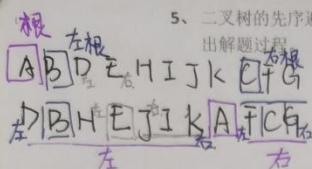
3、请画出图示二叉树的二叉链表存储结构。



4、请给出上图所示二叉树的前序、中序、后序、层次遍历序列。

前序遍历：A B D F C E G H
 中序遍历：B F D A G E H C
 后序遍历：F D B G H E C A
 层次遍历：A B C D E F G H

5、二叉树的先序遍历为：ABDEHJKCFG，中序遍历为：DBHEJKAFCG，请画出该二叉树。给



6、分析如下代码功能，并写出求二叉树中叶子结点的算法。

```
int Size(BiTree T)
```

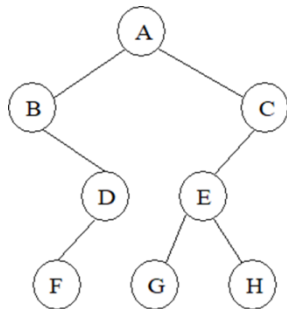
```

{
    if (T== NULL)
    { return 0; }
    else
    { return 1 + Size(T->Lchild) + Size(T->Rchild); }
}

int count(Bitree T)
{
    if (T == NULL)
    {
        return 0;
    }
    else if (T->Lchild==NULL && T->Rchild==NULL)
    {
        return 1;
    }
    else
    {
        return (count(T->Lchild) + count(T->Rchild));
    }
}

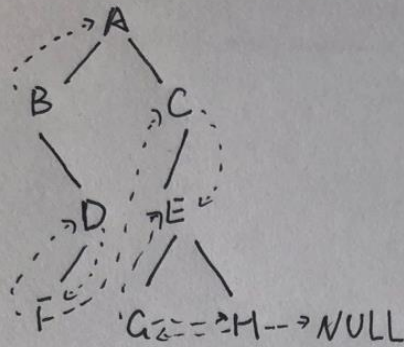
```

7、请画出如下二叉树前序线索二叉树、中序线索二叉树和后序线索二叉树。



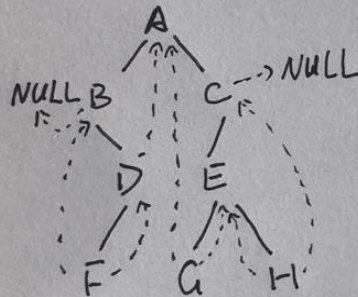
先序线索二叉树

先序序列: ABDFCEGH



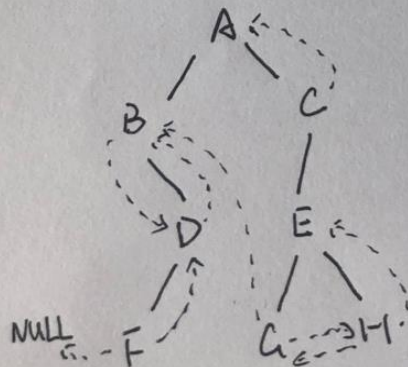
中序线索二叉树

中序序列: B A G E H C
FD



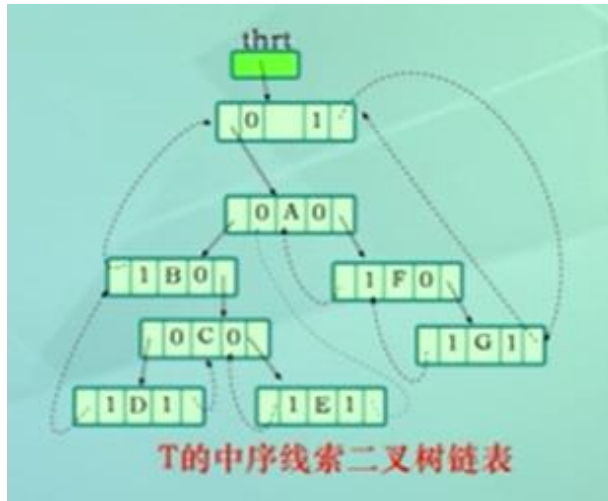
后序线索二叉树

后序序列: F D B G H E C A

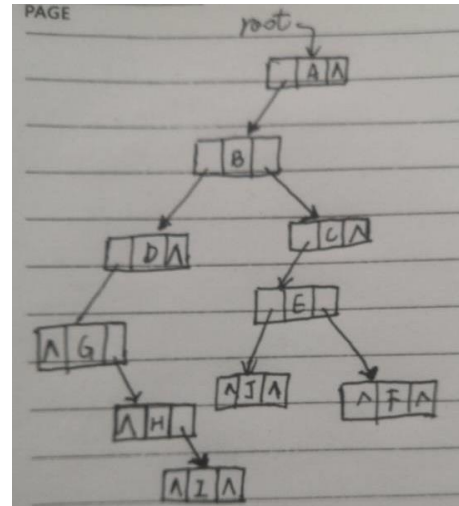
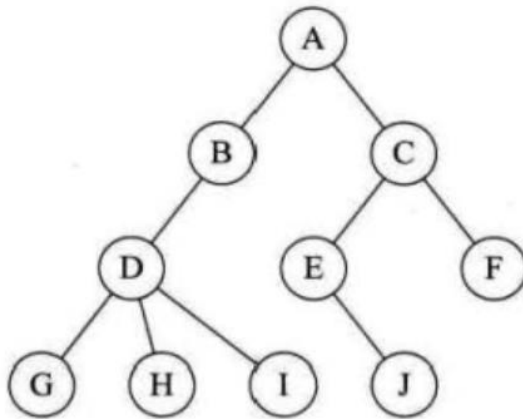


8、在线索二叉树中，如何查找结点的后继，如下图中序线索二叉树中，怎么找 D 的中序后继结点，怎么找 B 的中序后继结点？

D 无右子树，它的中序后继结点为后继线索所指结点即 A；B 有右子树，它的中序后继结点为对其右子树进行中序遍历时访问的第一个结点，即 D



9、给出图示树的孩子兄弟存储结构，并给出在孩子兄弟存储结构中，叶子节点的判断条件



孩子指针为空的结点为叶子结点

10、下图为4棵树构成的森林，完成下述问题：（1）给出该森林的先序遍历序列和中序遍历序列。（2）将森林转化为二叉树。（3）给出转化后二叉树的先序遍历序列、中序遍历序列、后序遍历序列。（4）比较森林的遍历序列和二叉树的遍历序列，是否有相同的序列。



11) 先序遍历序列
 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 15, 14

中序遍历序列
 3, 4, 8, 6, 7, 5, 2, 1, 10, 9, 11, 15, 13, 14, 12

12)

二叉树

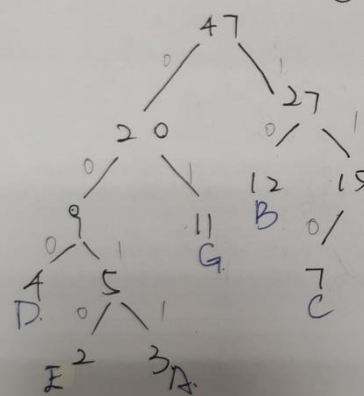
13) 先序遍历: 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 15, 14
 中序遍历: 3, 4, 8, 6, 7, 5, 2, 1, 10, 9, 15, 13, 14, 12
 后序遍历: 8, 7, 6, 5, 4, 3, 2, 10, 15, 14, 13, 12, 11, 9, 1

14) 先序遍历与中序遍历相同

- 11、 已知字符 A、B、C、D、E、F、G 的权值分别为 3、12、7、4、2、8、11，完成下列任务：（1）构造哈夫曼树；（2）为 7 个字母设计哈夫曼编码；（3）求该哈夫曼树的带权路径长度；（4）根据给出的哈夫曼树存储结构初态，继续填写，给出应哈夫曼树的存储结构终态。

前结构终态。

(1) $\begin{matrix} 3 & 12 & 7 & 4 & 2 & 8 & 11 \\ \textcircled{A} & \textcircled{B} & \textcircled{C} & \textcircled{D} & \textcircled{E} & \textcircled{F} & \textcircled{G} \end{matrix}$



(2) $W = 4 \times 3 + 2 \times 4 + 3 \times 4 + 11 \times 2 + 12 \times 2 + 7 \times 3 + 8 \times 3 = 123$

(4) 初始: 终态:

n	权值	父	左孩	右孩
1	3	3	0	8
2	12	12	0	12
3	7	7	0	10
4	4	4	0	9
5	2	2	0	8
6	8	8	0	10
7	11	11	0	11
8	0	5	0	9
9	0	9	0	11
10	0	15	0	12
11	0	20	0	13
12	0	27	0	13
13	0	47	0	0

(2) A: 0 0 1 1
 B: 1 0
 C: 1 1 0
 D: 0 0 0
 E: 0 0 1 0
 F: 1 1 1
 G: 0 1