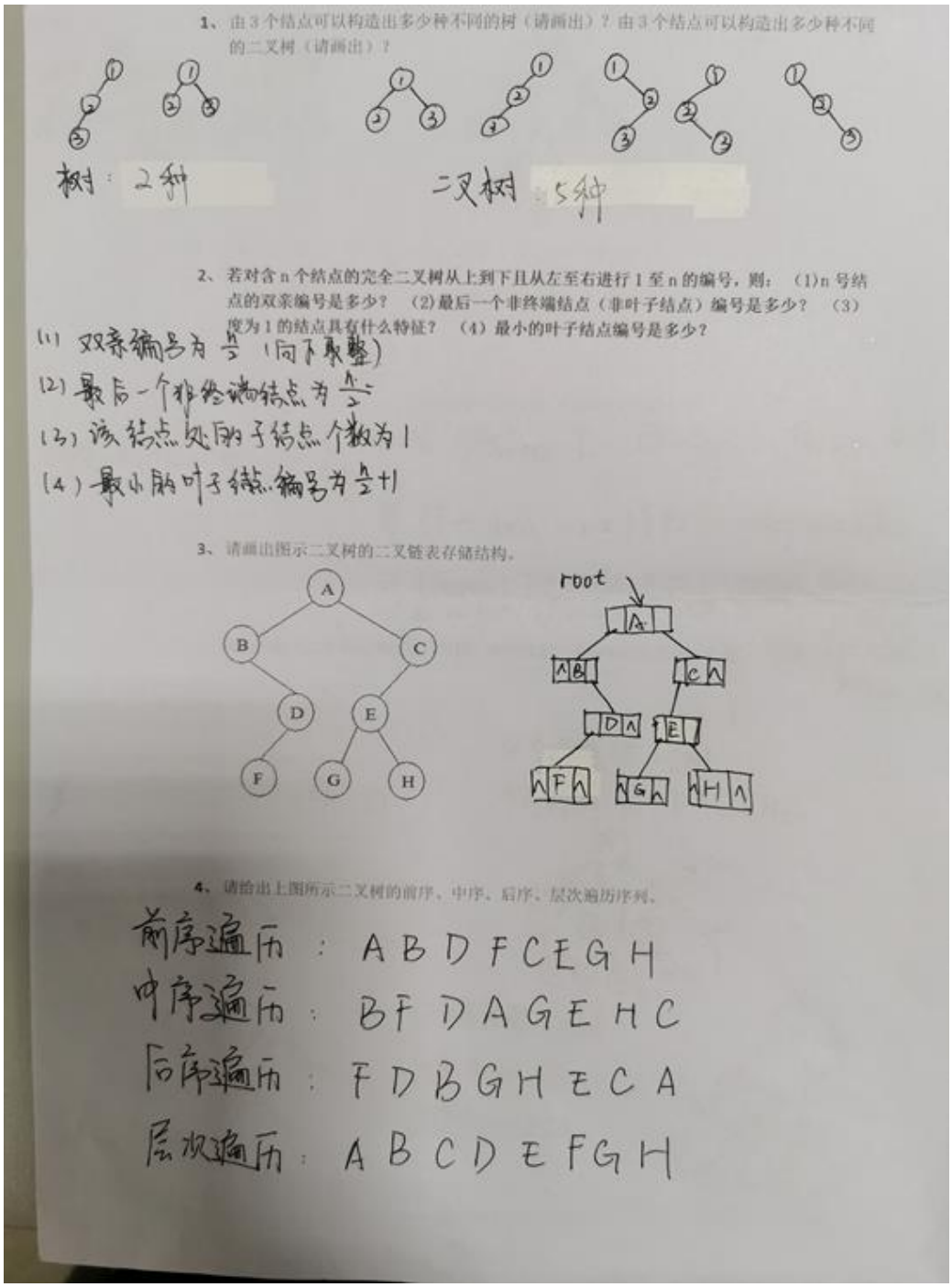
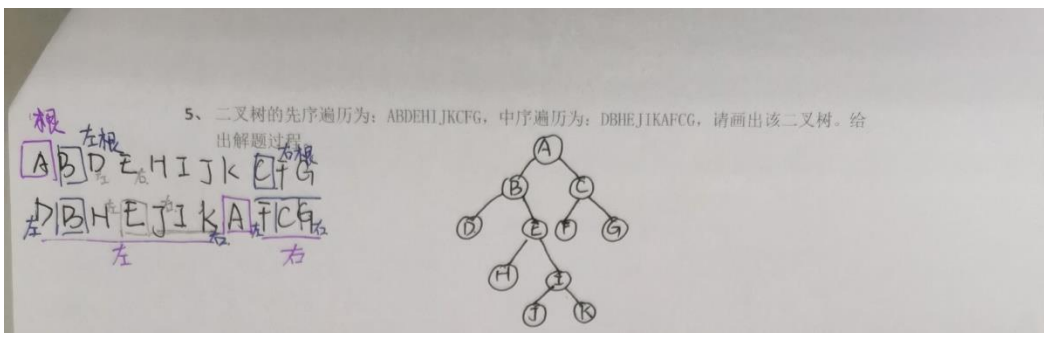



6、 分析如下代码功能，并写出求二叉树中叶子结点的算法。

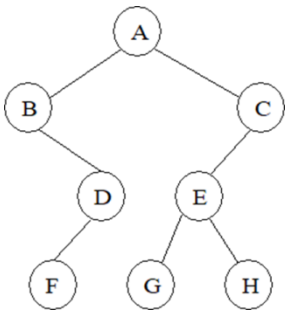
```
int Size(BiTree T)
```

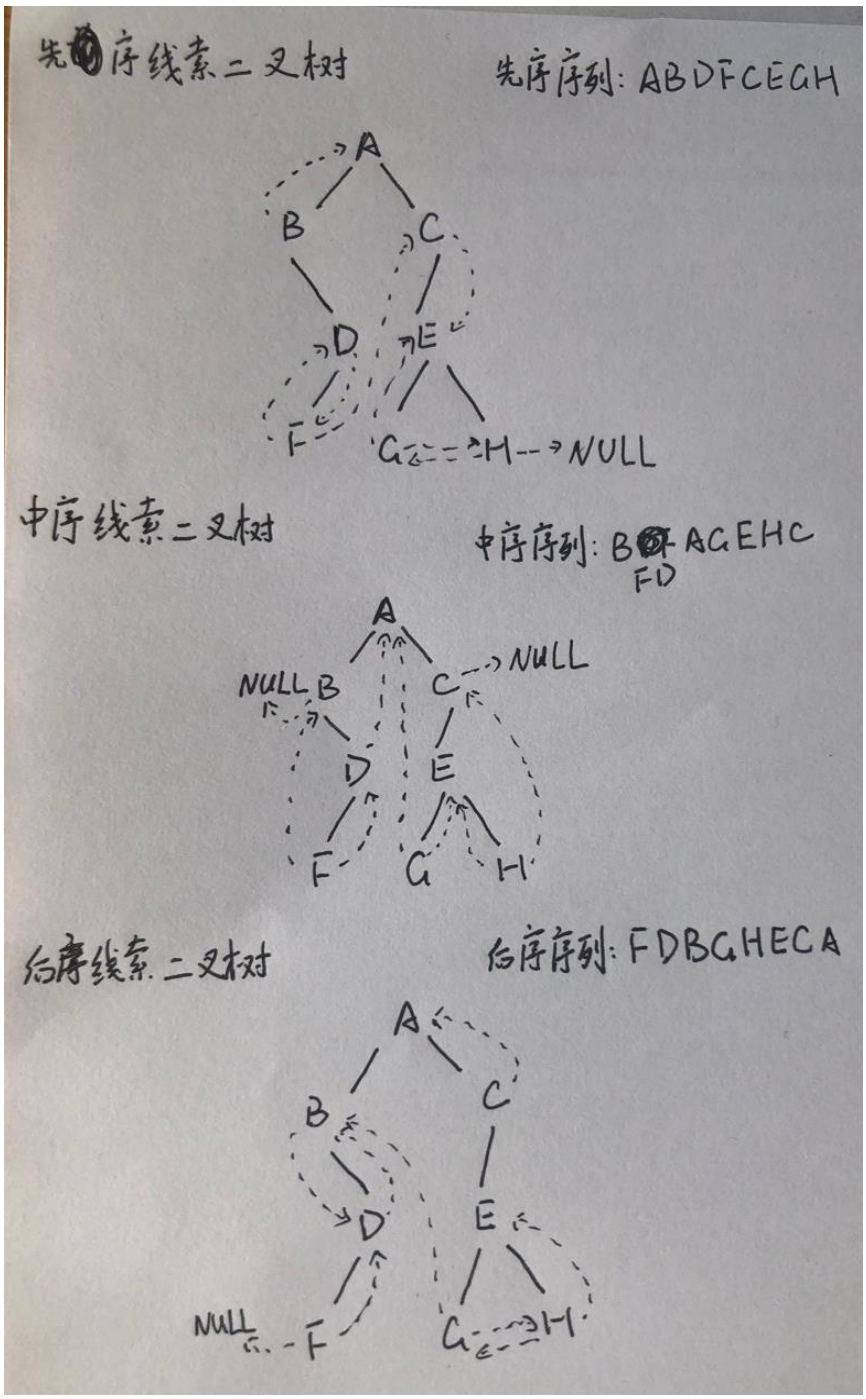
```
{
    if (T== NULL)
    {   return 0; }
    else
    {   return 1 + Size(T->Lchild) + Size(T->Rchild); }
```

```
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));
    }
}
```

```
}
```

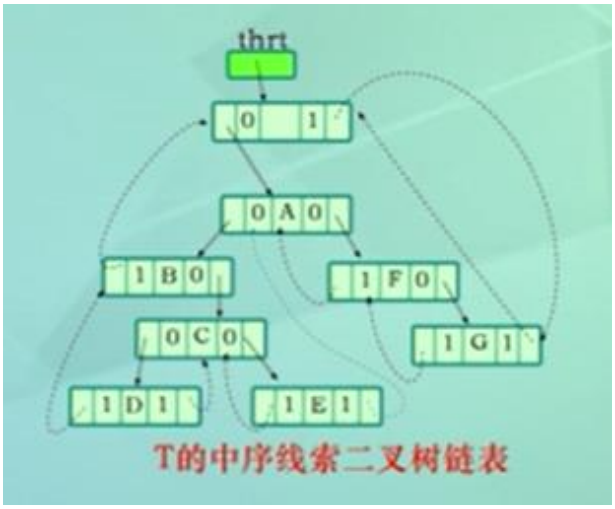
7、 请画出如下二叉树前序线索二叉树、中序线索二叉树和后序线索二叉树。



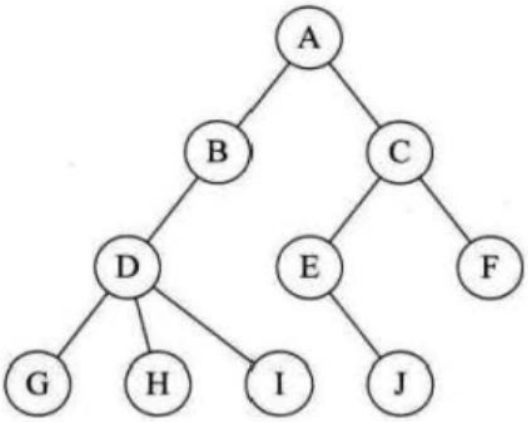


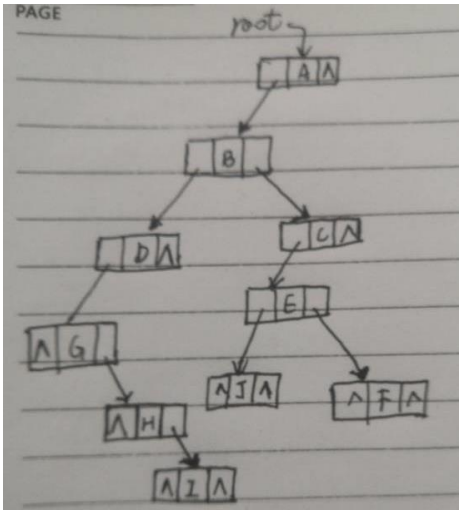

8、 在线索二叉树中，如何查找结点的后继，如下图中序线索二叉树中，怎么找 D 的中序后继结点，怎么找 B 的中序后继结点？

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



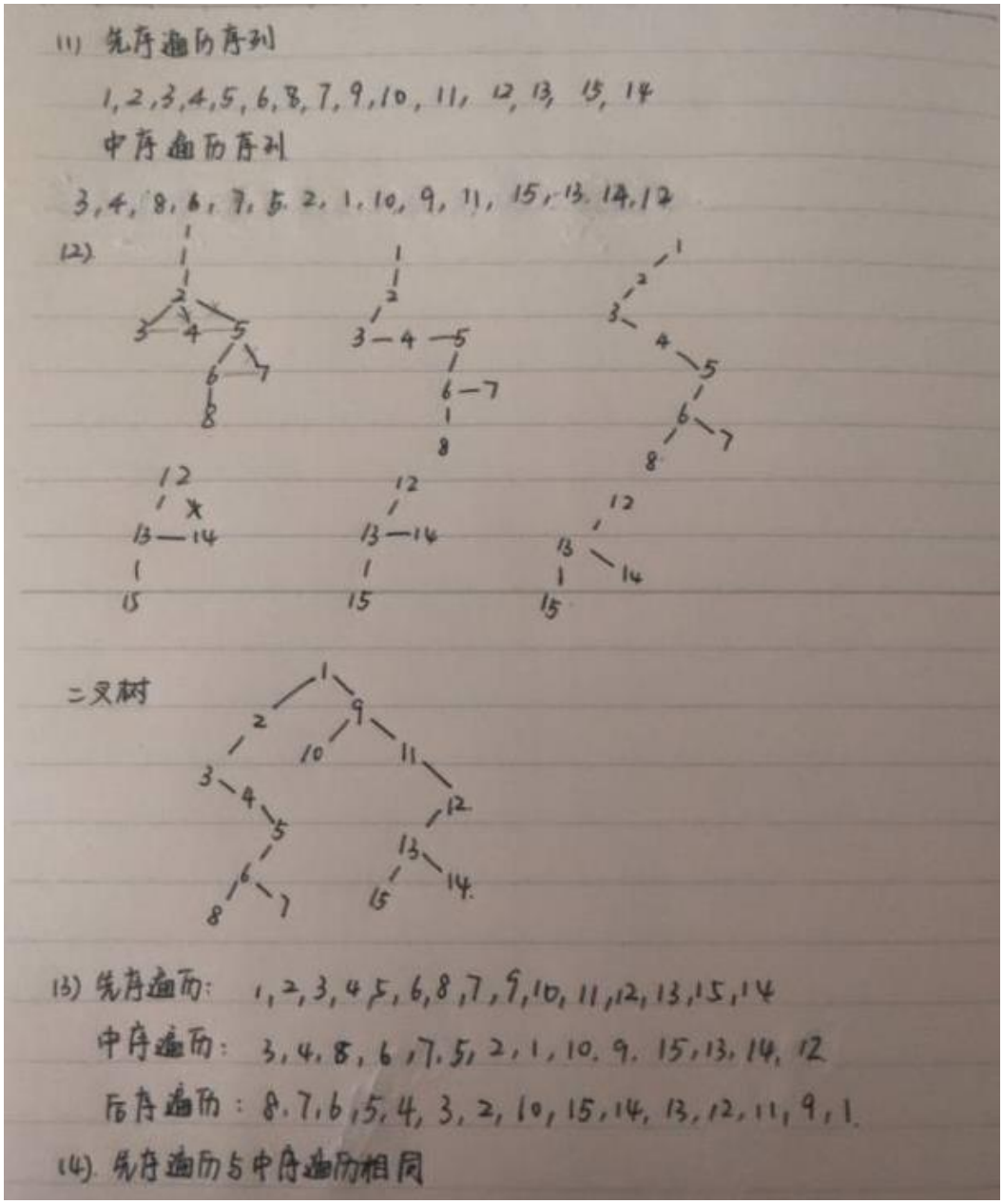

9、 给出图示树的孩子兄弟存储结构，并给出在孩子兄弟存储结构中，叶子节点的判断条件




孩子指针为空的结点为叶子结点

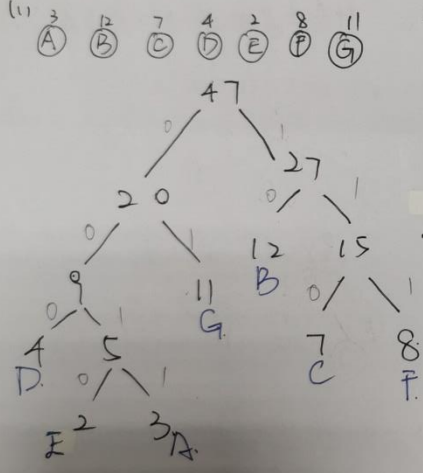
10、 下图为 4 棵树构成的森林，完成下述问题：（1）给出该森林的先序遍历序列和中序遍历序列。（2）将森林转化为二叉树。（3）给出转化后二叉树的先序遍历序列、中序遍历序列、后序遍历序列。（4）比较森林的遍历序列和二叉树的遍历序列，是否有相同的序列。




11、 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。

储结构终态。

(1)




(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

(3) $W = 4\times3 + 2\times4 + 3\times4 + 11\times2 + 12\times2 + 7\times3 + 8\times3$

$= 123$

(4) 初态：终态：

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