# 习题3 树结构

3-1设二叉树T中度为1的结点11个，度为2的结点12个，则二叉树T共有(   C  )个叶子结点。

(A) 11

(B) 12

(C) 13

(D) 36

3-2 设树T的度为4，其中度为1，2，3和4的结点个数分别为4，2，1，1，则T中的叶子数为( D )。

(A) 5

(B) 6

(C) 7

(D) 8

3-3 已知一棵度为k的树中，有n1个度为1的结点，n2个度为2的结点，…，nk个度为k的结点。试计算该树的叶子结点数。

答：设叶子结点数为，总结点数为。根据树的结点数与分支数的关系：N = n0 + n1 + ...... + nk，且树的分支数B = n1 + 2\*n2 + ......k\*nk，又因为N=B+1。所以n0=1+n2+2\*n3+......(k-1)nk。

3-4 证明：如果二叉树T的叶子结点数为n0，度为2的结点数为n2，则n0=n2+1。

证明：设二叉树的总结点数为，度为 0 的结点数即叶子结点数为n0，度为 1 的结点数为n1，度为 2 的结点数为n2，则n=n0+n1+n2。二叉树的分支数e，因为度为 1 的结点有 1 条分支，度为 2 的结点有 2 条分支，所以e=n1+2\*n2。又因为二叉树中分支数e=n-1，化简可得n0=n2+1。

3-5 对于任意非空二叉树，要设计出其后序遍历的非递归算法而不使用栈结构，最适合的方法是对该二叉树采用( B )存储结构。

(A) 二叉链表

(B) 三叉链表

(C) 索引

(D) 顺序

3-6 一棵二叉树的叶子结点在其先序、中序和后序序列中的相对位置( C )。

(A) 肯定发生变化

(B) 可能发生变化

(C) 不会发生变化

(D) 无法确定

3-7 设二叉树T按照二叉链表存储，则下列递归算法的主要功能是(   B  )。

int F(BiTree T)

{

if (!T) return 0;

x=F(T->Lchild);

y=F(T->Rchild);

if (y>x) x=y;

return x+1;

}

(A) 交换二叉树T的左右子树

(B) 计算二叉树T的高度

(C) 计算二叉树T的叶子结点数

(D) 先遍历左子树，再遍历右子树

3-8 已知二叉树T的先序序列为ABCDEF，中序序列为CBAEDF, 则T的后序序列为( C )。

(A) CBEFDA

(B) FEDCBA

(C) CBEDFA

(D) 不确定

3-9 简述由先序序列和中序序列构造二叉树的基本操作方法。

答：（1）先序序列的第一个元素就是二叉树的根结点。

（2）在中序序列中找到根结点的位置，根结点左边的元素构成左子树的中序序列，右边的元素构成右子树的中序序列。

（3）根据左子树的中序序列长度在先序序列中确定左子树的先序序列（除去根结点后的前面部分），同理确定右子树的先序序列。

（4）分别对左子树和右子树的先序序列和中序序列重复上述步骤，递归构造出整个二叉树。

3-10 已知二叉树的先序序列为ebadcfhgjik，中序序列为abcdefghijk，试画出该二叉树。

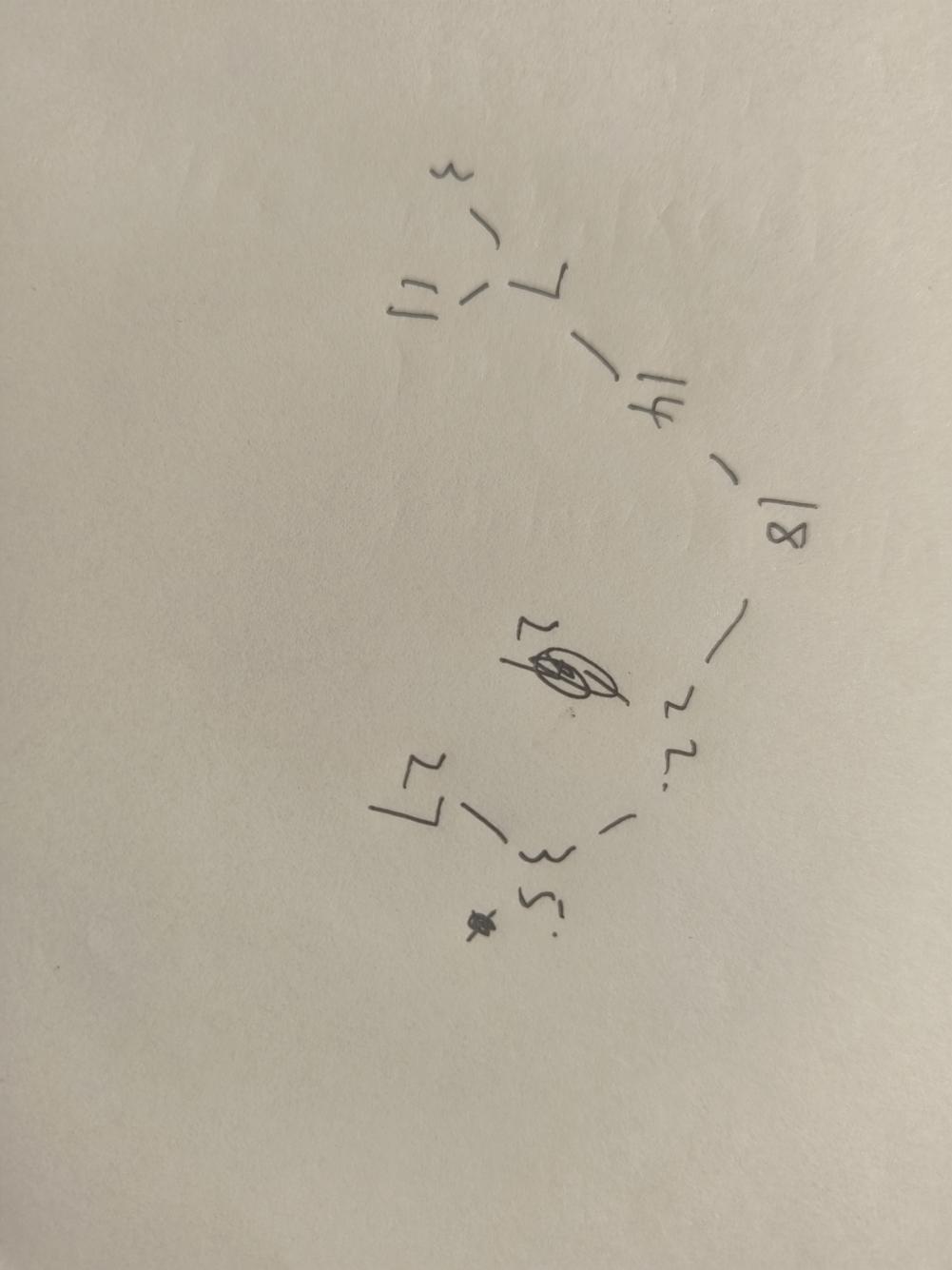
答：

3-11 已知二叉树T的中序序列和后序序列分别为

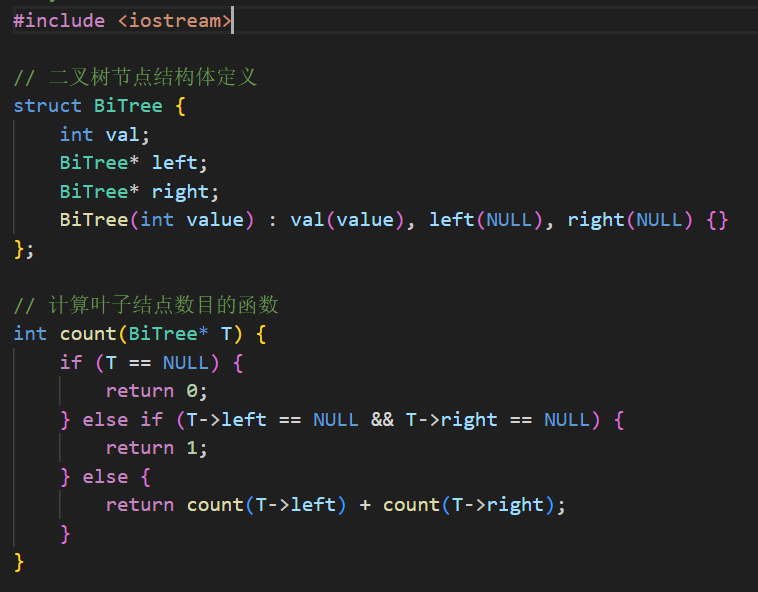
(中序) 3, 7, 11, 14, 18, 22, 27, 35

(后序) 3, 11, 7, 14, 27, 35, 22, 18

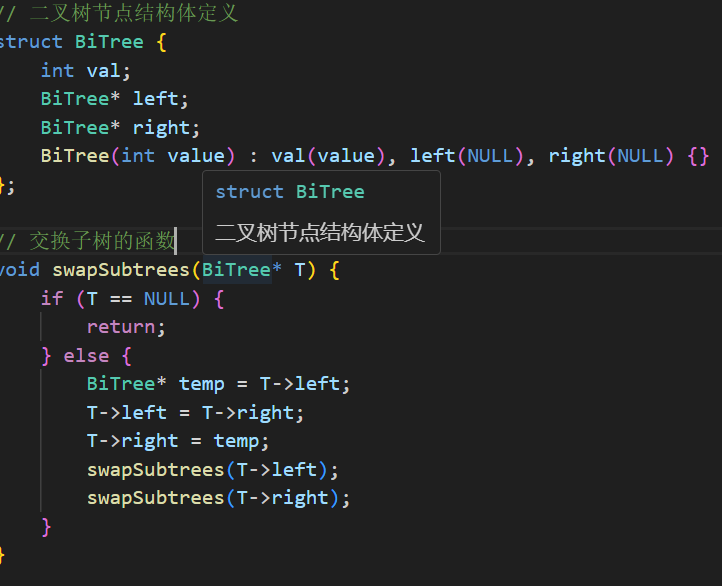
试画出二叉树T。



3-12 已知二叉树T按照二叉链表存储，设计算法，计算T中叶子结点的数目。



3-13 已知二叉树T按照二叉链表存储，设计算法，交换T的左子树和右子树。



3-14 先序后继线索化算法是根据二叉链表建立先序后继线索二叉链表，其基本原则是在前驱空指针域中写入后继线索，即将右子树的( D )指针写入左子树的最后一个叶子结点右指针域。

(A) 线索

(B) 根结点

(C) 前驱结点

(D) 后继结点

3-15 设计算法，在先序线索二叉树中，查找给定结点p在先序序列中的后继。



3-16对n (n≥2)个权值均不相同的字符构造哈夫曼树T，不正确的叙述( A )。

(A) T一定是一棵完全二叉树

(B) T中一定没有度为1的结点

(C) T中两个权值最小的结点一定是兄弟结点

(D) T中任一分支结点的权值一定不小于下一层任一结点的权值

3-17设计一个求结点x在二叉树中的双亲结点算法。

