# 6.1程序思路

1. 中序和后序依然可以递归方法构造树，与中序的主要区别在于对左右子树的指针移动规则不同，并且中序和后序的变化方式也不一样：后序每次分割后都取走末尾元素作为根节点，这和先序是相反的。
2. 计算二叉树高度也可以使用递归来完成，每次递归调用的时候，有子树就对高度加1即可，无子树就返回0，不增加高度。
3. 计算叶子节点使用先序遍历方法，遇到无左右子树的即认为是叶子节点，对计数器num+1即可。

# 6.2程序思路

1. 使用字符指针数组存储霍夫曼编码。
2. 每次使用Select函数在输入的字符集中找到最小的两个值并保存到s1，s2变量中
3. 实际上是用数组来构建霍夫曼树，因为链表的查询时间太慢，而且C语言没有封装好的高级数据结构，如果使用哈希将提高查找最小两个节点的速度，为O（1）
4. 生成霍夫曼编码的时候使用倒序，先指向 ’\0’ 字符
5. 假如使用链表进行操作，可以使用先序遍历，进入左子树则为0，右子树为1

# 6.2debug反馈

原先代码就是根据需要为子树分配空间，无需更改。