##### 实验报告（五）

实验内容

**Part 1：**

用递归的方法实现以下算法：

1．以二叉链表表示二叉树，建立一棵二叉树；

2．输出二叉树的中序遍历结果；

3．输出二叉树的前序遍历结果；

4．输出二叉树的后序遍历结果；

5．计算二叉树的深度；

6．统计二叉树的结点个数；

7．统计二叉树的叶结点个数；

8．统计二叉树的度为 1 的结点个数；

9．输出二叉树中从每个叶子结点到根结点的路径。

提高内容

1．交换二叉树每个结点的左孩子和右孩子；

2．设计二叉树的双序遍历(DblOrderTraverse)算法（双序遍历是指对于二叉树的每一个结点来说，先访问这个结点，再按双序遍历它的左子树，然后再一次访问这个结点，接下来按双序遍历它的右子树）。

**Part 2：**

实现一个哈夫曼编码系统，系统包括以下功能：

(1) 字符信息统计：读取待编码的源文件 SourceFile.txt，统计出现的字符及其频率。

(2) 建立哈夫曼树：根据统计结果建立哈夫曼树。

(3) 建立哈夫曼码表：利用得到的哈夫曼树，将各字符对应的编码表保存在文件 Code.txt 中。

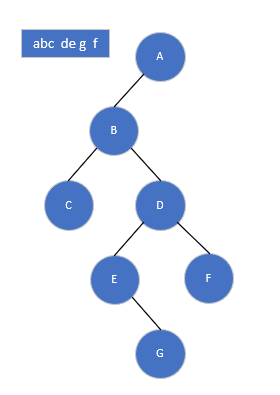
(4) 对源文件进行编码：根据哈夫曼码表，将 SourceFile.txt 中的字符转换成相应的编码 文件 ResultFile.txt。

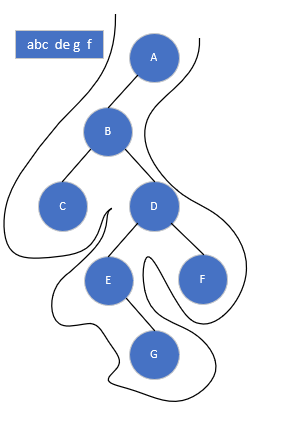
提高内容:

完成译码功能：对任意一个给定的由 01 组成的文件，根据哈夫曼码表翻译成由字符 组成的源文件。

主要算法流程图

**Part1:**





按照如图顺序，遇见一个点有三次，

第一次就输出是先序遍历

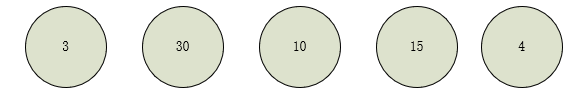
第二次输出是中序遍历

第三次输出是后序遍历

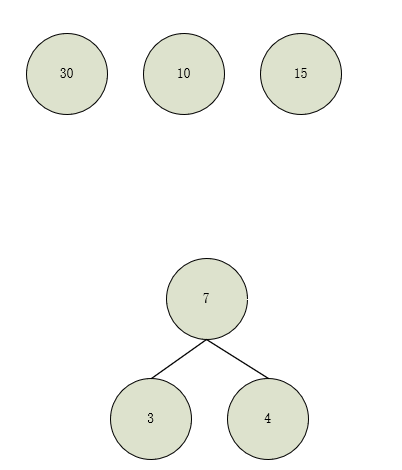
**Part2:**

**Huffman树的构建，依次选择最小的两个子树合并**

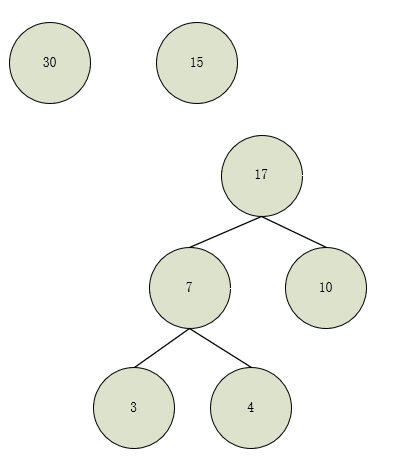
1.



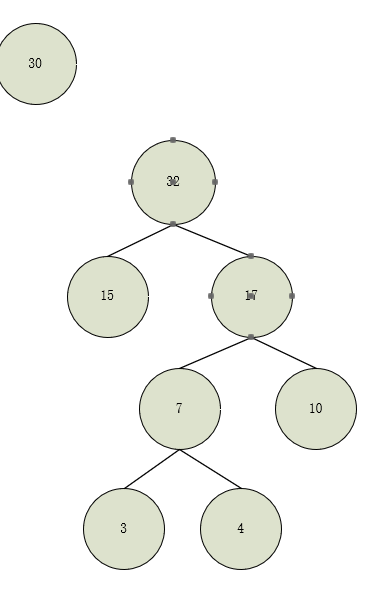
2.



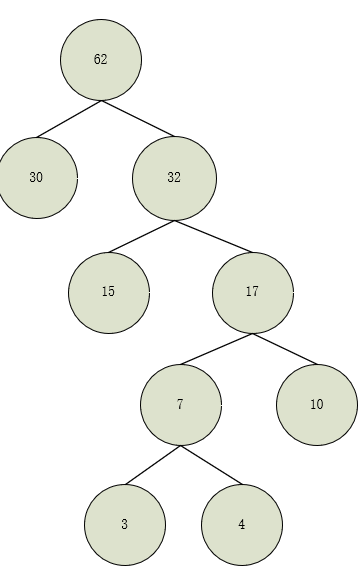
3.



4.

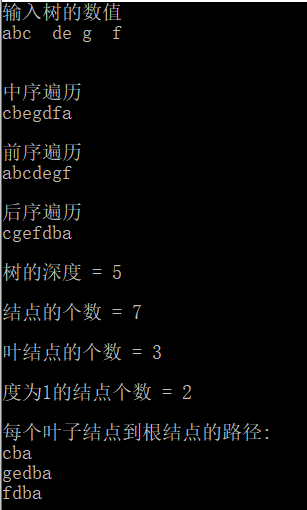


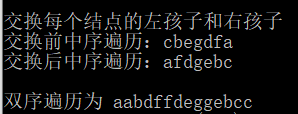
5.



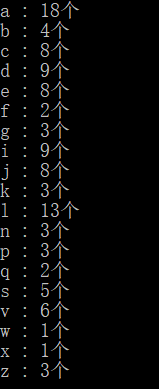
实验结果

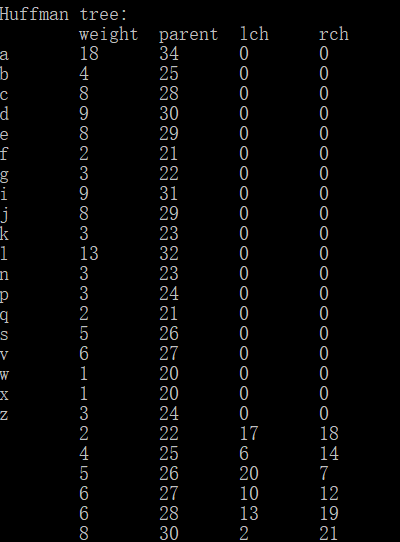
**Part 1：**

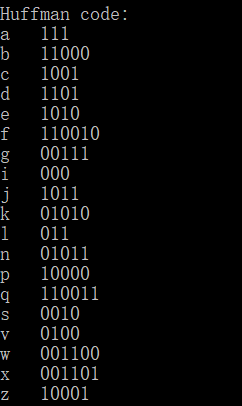


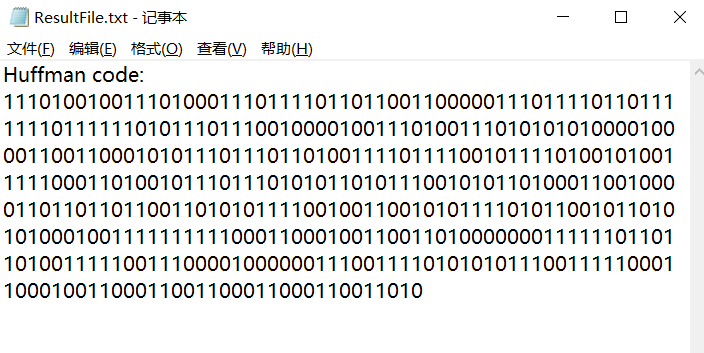


**Part 2：**











实验小结

1.学会熟练使用递归遍历树

2.学会二叉树的建立方法

3.学会了保存路径来输出每个叶结点到根结点的路径

4.学会了哈夫曼树的建立方法

5.学会了哈夫曼编码的创建方法

6.学会了通过哈夫曼树递归译码的方法