实验报告

## 实验名称 ：赫夫曼编译码系统

## 需求分析

题目为：写 一个完整的系统应具备一下功能

（1）I:初始化（Initialization）。从终端读入字符集大小n，以及n个字符和n个权值，建立哈夫曼树，并将它存于文件hfmTree中。

（2）E:编码（Encoding）。利用已建好的哈夫曼树（如不在内存，则从文件hfmTree中读入），对文件ToBeTran中的正文进行编码，然后将结果存入文件CodeFile中。

（3）D:译码（Decoding）。利用已建好的哈夫曼树将文件CodeFile中的代码进行译码，结果存入文件TextFile中。

（4）P:打印代码文件（Print）。将文件CodeFile以紧凑格式显示在终端上，每行50个代码。同时将此字符形式的编码文件写入文件CodePrin中。

（5）T:打印哈夫曼树（Tree printing）。将已在内存中的哈夫曼树以直观的方式（树或凹入表形式）显示在终端上，同时将此字符形式的哈夫曼树写入文件TreePrint中。

分析：题目的意思已经很清楚了，就是写一个程序来实现这5个功能。所以，这里不做赘述。

## 概要设计

1. 总体设计：main函数里进行与用户的交互，每一个功能为一个函数，当用户选择某一功能的时候，调用特定的函数完成功能即可.

2.对(1)进行分析可得:

需要建立哈夫曼树，并将树的节点信息存储到文件hfmTree中,那就设计一个函数交init()此函数完成建立哈夫曼树，并将节点信息存储到文件中:

3.对(2)分析:

先读取文件hfmtree，若不存在则告诉用户进行输入数据，如果存在，读入哈夫曼树并对文字进行编码,最后将编码的结果存到CodeFile中即可

4.对(3)分析:

用已建好的哈夫曼树将文件CodeFile中的代码进行译码，结果存入文件TextFile中。

5.对(4)分析:

将文件CodeFile以紧凑格式显示在终端上，每行50个代码。同时将此字符形式的编码文件写入文件CodePrin中

6.对(5)分析:

将已在内存中的哈夫曼树以凹入表形式显示在终端上，同时将此字符形式的哈夫曼树写入文件TreePrint中。

## 详细设计.

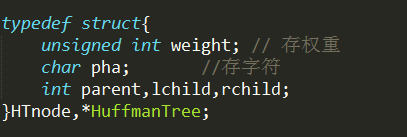
1. 总体设计：main函数里进行与用户的交互，每一个功能为一个函数，当用户选择某一功能的时候，调用特定的函数完成功能即可.效果如下:



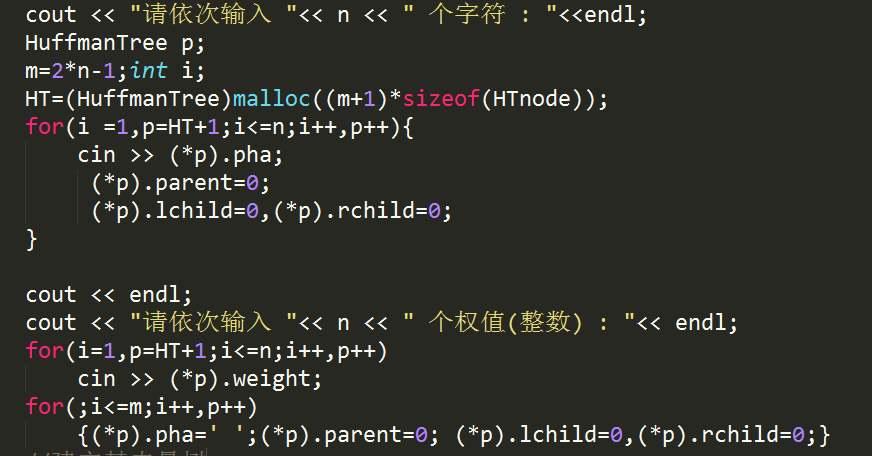
2. 对(1)进行分析可得:

需要建立哈夫曼树，并将树的节点信息存储到文件hfmTree中,那就设计一个函数交init()此函数完成建立哈夫曼树，并将节点信息存储到文件中:

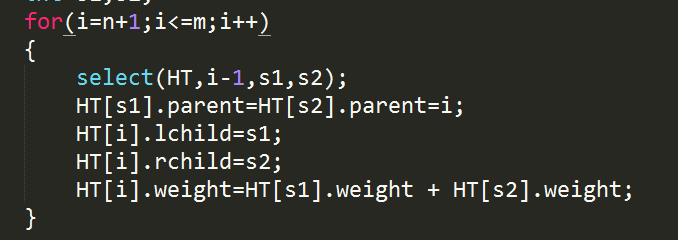
首先需要建树，那么存储结构如何选择？我选择了顺序储，每一个节点存储权值以及字符，并存储它的左右子树和双亲节点的下标，结构如图:

1. 

建树思路为: 首先由用户输入字符的个数，并根据字符的个数申请上述的一个结构数组,大小为2\*n个,并一次将parent,lchild,rchild置为0，同时由用户输入n个字母，并顺序的存放在前n个位置(如下图所示)，接下来再要求用户输入权值,然后开始建树.

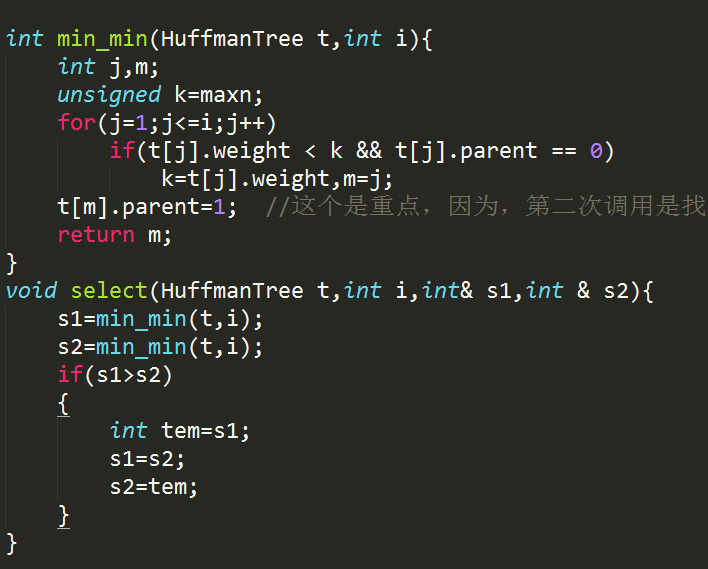


首先从所以的已知的节点中挑出最小的两个结点，新节点从第n+1个开始每挑出两个节点s1,s2，就将新节点的权值置为s1和s2的权值和,并将s1和s2的父亲节点设为第n+1个节点，并将他的左右孩子分别设为s1，和s2



建树过程如图，具体的怎么挑权值最小的两个结点呢？

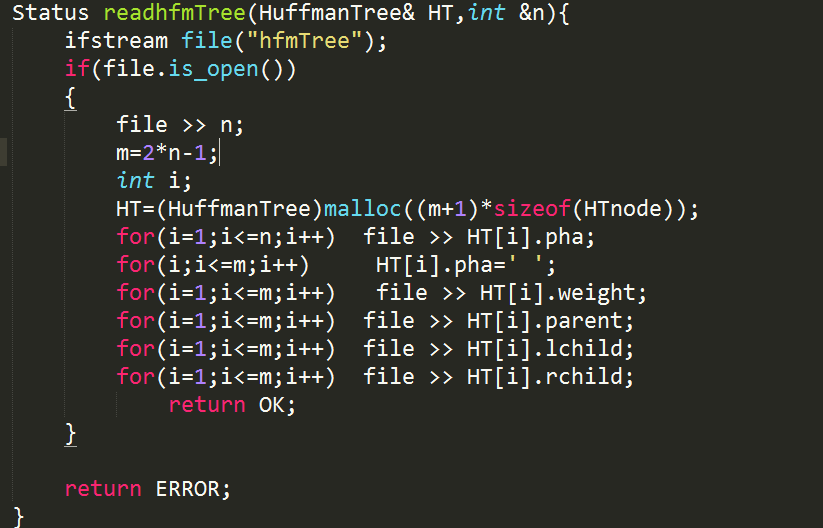
首先，遍历一次所以的节点，找到最小结点的小标，并将它的双亲节点置为1(这里改了之后不会对后来造成影响，由于之后要更改它的parent为新加的结点，故此处这么设置并无影响)，那么第二次就能挑出次小的结点。(具体如下图)



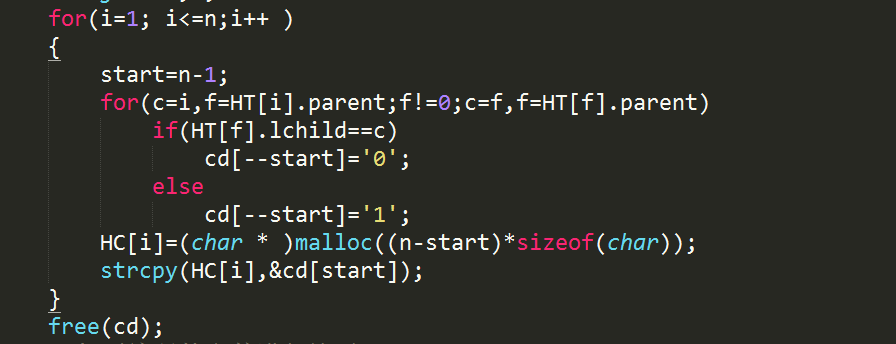
由于需要将哈夫曼树存入到文件中，所以调用c++中的文件输出流，将结点数组输出即可。

3.对(2)分析:

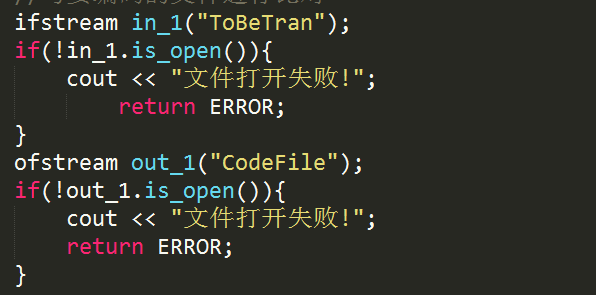
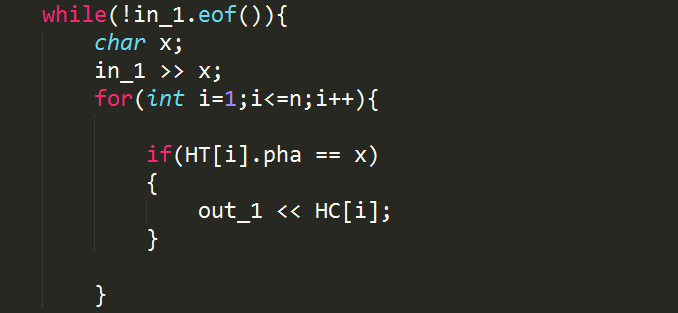
先读取文件hfmtree，若不存在则告诉用户进行输入数据，这部分可以由c++中的文件操作函数file.is\_open()的返回值来确定是否存在，读入数据直接保持在HT[]中即可。



如果存在，读入哈夫曼树并进行编码，即对树中的每一个元素进行编码，并将编码的结果存到一个字符数组HC中，中间借助了辅助数组cd由于是从叶子节点到根节点的。编码顺序是逆向的，只能如次了，具体如图:



之后打开文件ToBeTran和CodeFile，根据ToBeTran读入的没一个字符，在HC中比对得到其编码输出到CodeFile中即可,具体如图:

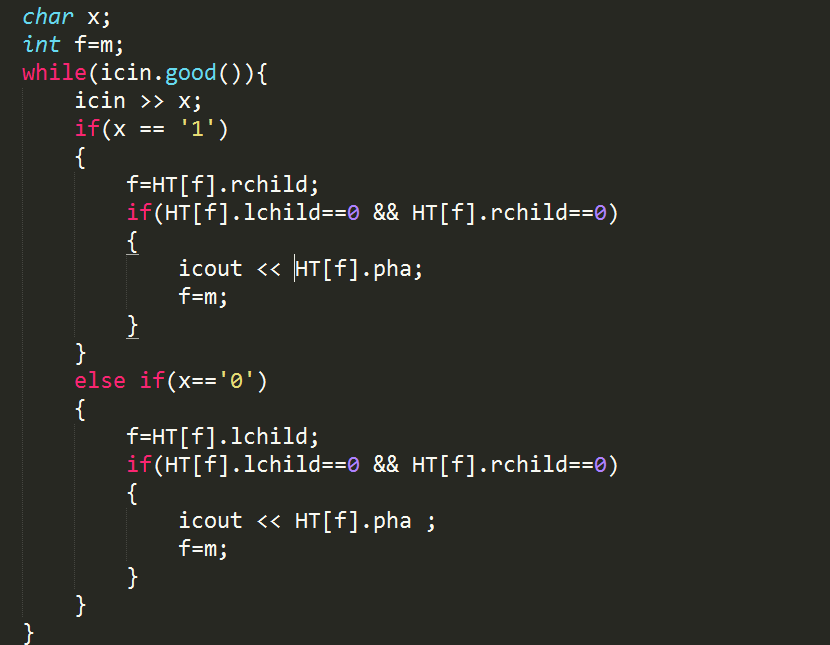
  
4.对(3)分析:

用已建好的哈夫曼树HT将文件CodeFile中的代码进行译码，

具体做法为:

由于CodeFile中寸的是0和1这就有一个很好的作用，我直接可以从根节点出发，遇到1就到rchild，遇到0就到lchild。

到叶子节点直接就将叶子节点的字母输出到TextFile中。(具体过程如图)



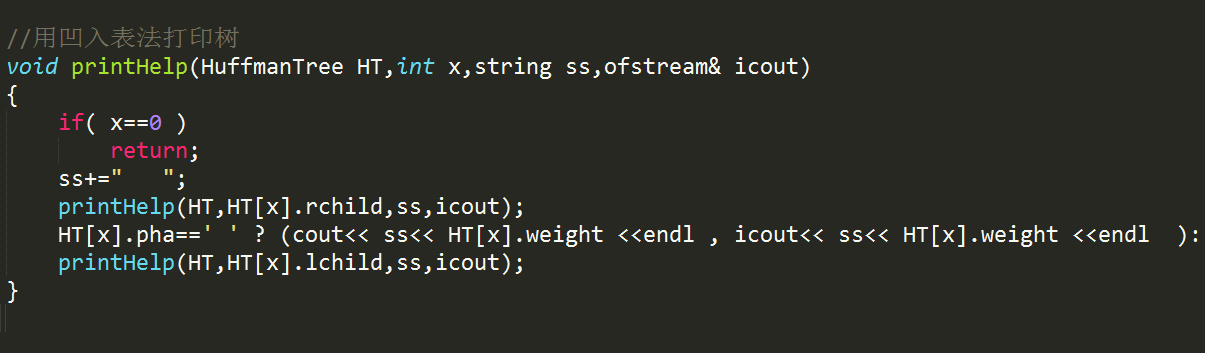
5.对(4)分析:

将文件CodeFile以紧凑格式显示在终端上，每行50个代码。这就是打开文件，并将字符输出到终端上,同时将此字符形式的编码文件写入文件CodePrin中同样的和在终端的操作一样。

6.对(5)分析:

将已在内存中的哈夫曼树以凹入表形式显示在终端上同时将此字符形式的哈夫曼树写入文件TreePrint中。

此部分，可以将TreePrint文件提前打开。输出是作为流输出即可。首先想到的是递归，利用字符串的可加性，没递归一层增加一个空格的长度，直到遇到叶子节点返回。具体如图：



## 用户手册

**测试数据：**

**8**

**A B C D E F G H**

**5 29 7 8 14 23 3 11**

**ToBeTran文件内容: ABCDEFGHABCCDD**

**26**

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

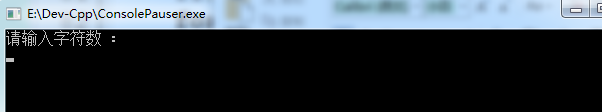
**64 13 22 32 103 21 15 47 57 1 5 32 20 57 63 15 1 48 51 80 23 8 18 1 16 1**

**程序的主界面如下:**

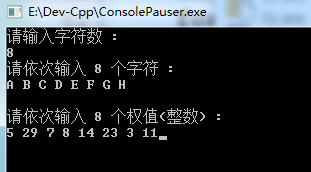


**首先选择 I :**

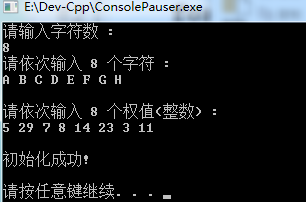
**界面如下:**



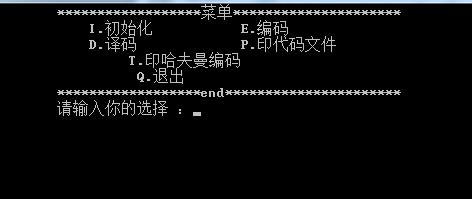
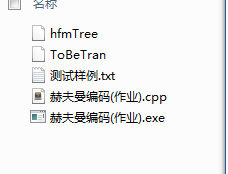
**输入上述的测试数据:**



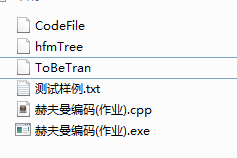
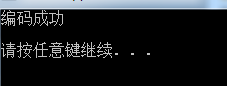
**此时回车，会出现如下的初始话成功的提示:**



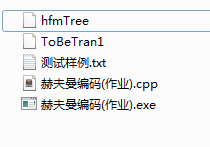
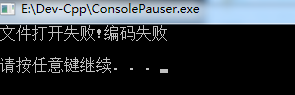
**此时按下任意键会返回到初始的菜单。并且同路径下多了一个hfmTree的文件如图：**



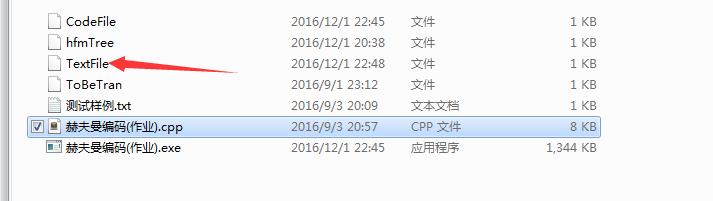
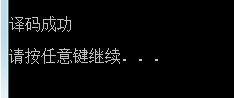
**接下来选择 E进行编码。 注意，此时你的程序运行路径下应该存在ToBeTran文件，内容可自行编辑，如果存在会出现以下提示并且路径下多了一个名叫CodeFile的文件如下：**

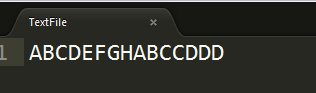


**若不存在ToBeTran。会有以下提示:**

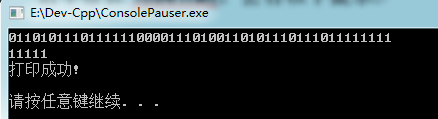


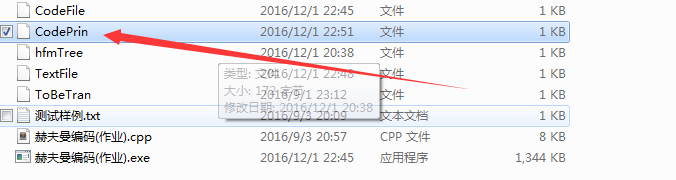
**接下来选择 D进行译码，如果成功会出现如下窗口，同样会创建一个TestFIle的文件 ：**

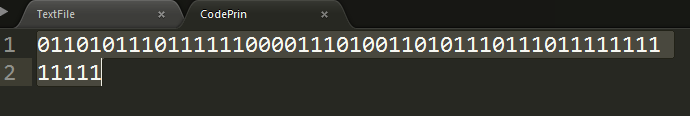


 **TextFile如左图所示**

**接下来输入 P 进行印代码出现如下 , 同时，目录下会现CodePrin文件及其内容:**

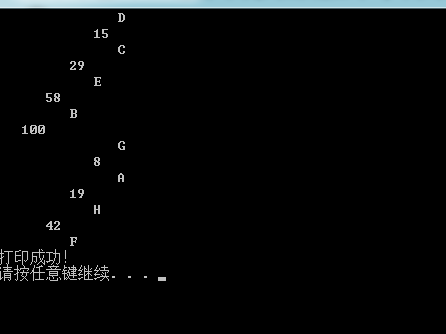


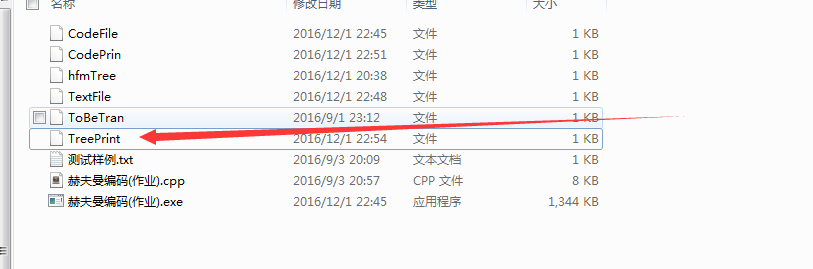




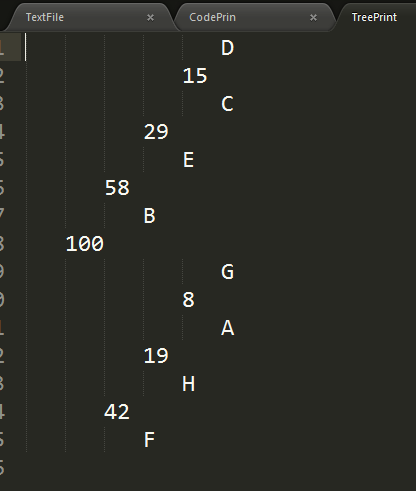
**接下来输入T：**

**打印出来了哈夫曼树，并且目录下创建了TreePrint文件**

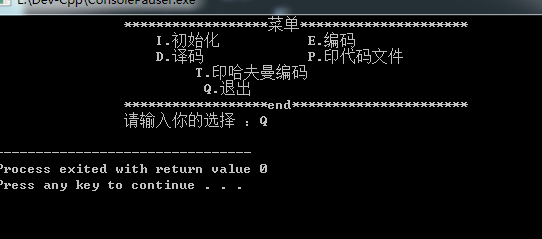




**TreePrint内容如下:**



**最后输入Q即可退出程！**



## 附录

输入数据文件名称: 测试样例.txt和ToBeTran

主要程序: 赫夫曼编码(作业).cpp

注：若出现乱码，请换一下编码即可。