## 图论实验三 制作文档

一、实验内容

从文件读入一段文本，计算其中每一个字符的huffman编码，输出编码后文本的长度。本程序以Windows平台下的Visual Studio 2012为开发环境，用C++语言实现上述功能。

二、实验要求

输入数据保存在文件input.txt 中，程序运行后将输出数据保存至**[**学号**].txt**中。这两个文件都与程序在相同的路径下。

输入文件包含一段文本，该文本可由各种ASCII字符组成（英文字母、标点、空格、换行等）。输出文件仅有一行，包含1个数字，表示使用 huffman编码后文本的长度。

三、设计思路

程序首先应能够读入文件中的所有字符，包括空格与换行符。通过定义ifstream对象inFile并通过inFile.get()逐个读取文件中的每一个ASCII字符。通过一遍读取得到文件中的总字符数后，通过new生成一个对应长度的数组FullFile储存所有字符。

第二步，程序应对文件中出现的所有字符进行计数。首先实现findChar函数与countChar函数分别实现查找字符与计数重复字符的功能。随后遍历一次FullFile即可完成第二步，出现过的字符无重复地存入数组diffChar，字符对应的出现次数存入数组diffCharCount。

随后实现HuffTree类的编写，完成对Huffman树的刻画。而creatTree函数利用queue中的优先队列完成构造HuffTree的过程。最后，getLen函数利用递归完成对编码后文件长度的计算，通过叶节点的层数乘以权值即为该叶节点对应的字符编码后的总长度。ASCII码数目小于256，递归深度不会超过8。

四、关键代码分析

第一个值得注意的代码段是getFileLen函数中对文件打开是否成功的判断与处理，如果打开失败，则输出“Could not open the file <input.txt>”并结束程序。

其次是main函数中判断文件中是否没有字符或是字符数为1，此时没有Huffman树的构造过程，一般计算将得出不正确的结果，因此特殊处理，向文件分别写入0和1即可。

其次是将出现过的字符无重复地存入数组diffChar以及字符对应的出现次数存入数组diffCharCount的过程。首先对FullFile中读取的字符在diffChar中查找，若已有则跳过。第二点应注意读取FullFile的序号与数组diffChar和diffCharCount的序号不同。

其次是HuffTree类的编写。该类由两个HuffTree指针，权值float Priority与该节点代表的字符char ch组成。两个指针分别代表左子节点与右子节点。无参数的构造函数初始化四者均为零，而重载的构造函数将该字符的出现次数作为该节点的Priority。

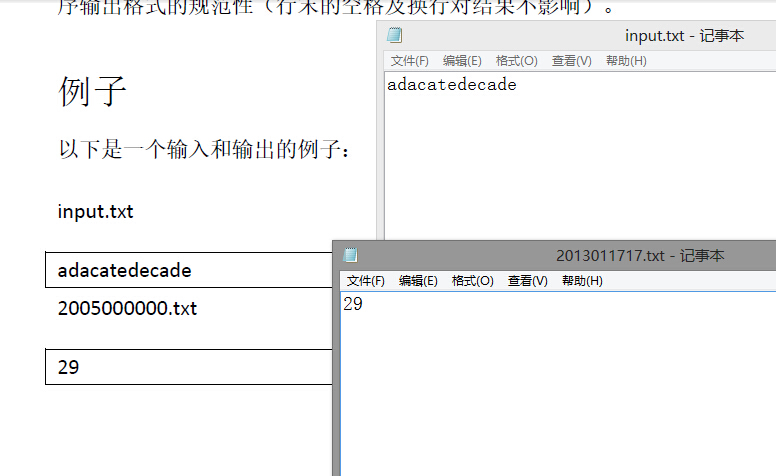
其次是在creatTree函数中使用优先队列。因对模版等较高级的C++技能了解程度有限，此处在网上进行了相关搜索对优先队列的使用方法进行了进一步学习，了解到了比较函数可用一个类给出，类中的函数重载了操作符()，可通过Priority比较两个HuffTree实例，优先队列中的元素则为HuffTree\*。有了优先队列后，HuffmanTree的构造便极为简单。取优先队列中的top赋给leftChild，pop后再取top赋给rightChild即为新构造的HuffTree节点的两个子节点，而新节点由new创建，并将相应指针赋值为leftChild与rightChild，将新节点的Priority赋为子节点的Priority之和，加入队列。如此循环直到队列中仅有一个元素为之，该节点即为根节点。

计算编码后文件长度的getLen函数，实现也较为简单。搜索到叶节点时返回该节点的Priority乘以该节点的深度即为该字符编码后在文件中所占的总长度。而该节点不为叶节点时返回两个子节点的getLen值之和即可。

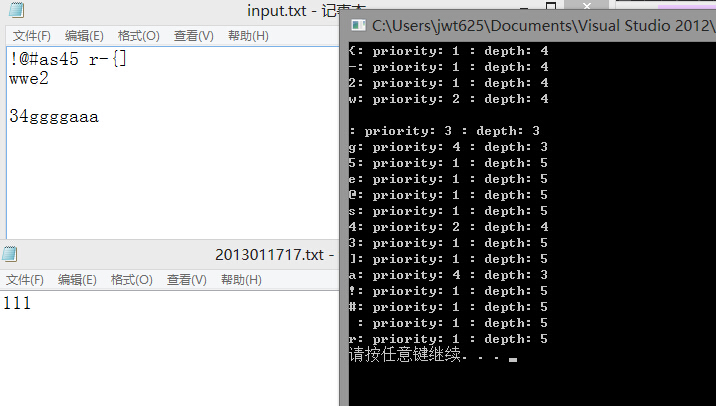
文件输出部分，程序运用C++已有的fstream，将最后得到的编码后文件长度写入2013011717.txt中。最后，程序关闭文件2013011717.txt。

五、实验结果与分析

以下为样例输入与输出：

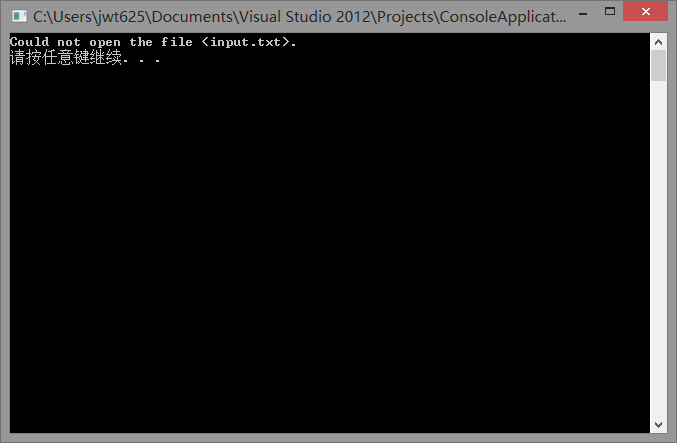


输出与所给输出一致。下面为另一组输入与输出测试：



上图中略微增添了控制台的输出以方便查看Huffman Tree的叶节点的具体情况

下面为没有input.txt文件时程序的输出：



综上，程序在正确的输入下能够完成正确的输出，并能在读取文件失败时反馈给用户。

六、实验小结

本次实验要求从文件读入一段文本，计算其中每一个字符的huffman编码，输出编码后文本的长度。通过这次实验，我练习了对Huffman编码的具体实现，加深了对Huffman编码的理解，以及对其的具体的程序实现进行了实践与熟悉，对一个树的类进行了编写实现与相关操作，对C++的优先队列有了了解与具体的使用经验，对文件读取的方法也有了更多了解与使用经验。

不足的是由于编程经验较少，程序在不少地方仍可能有所缺陷与漏洞；在算法方面也还有许多地方有待改进，如用动态数组可能减慢运行速度。以上就是本次实验的小结。