# 大作业二：利用哈夫曼进行压缩和解压缩

1. 实验要求

对所有文件类型进行压缩，压缩后的文件后缀名为.huf。同时，可以对所有后缀名为huf的压缩文件进行解压缩。

1. 实验内容

利用哈夫曼的编码和译码来对所有类型的文件进行压缩，压缩后的文件后缀名为.huf，同时，对所有后缀名为.huf的压缩文件进行解压缩。

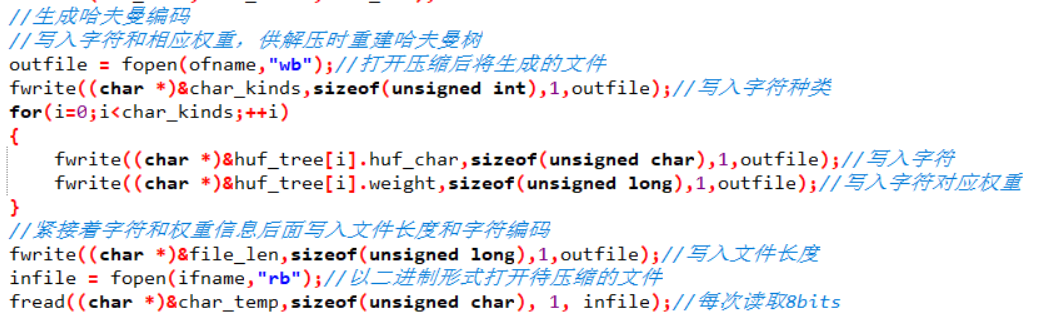
1. 关键代码讲述：

（1）.结构体的创建



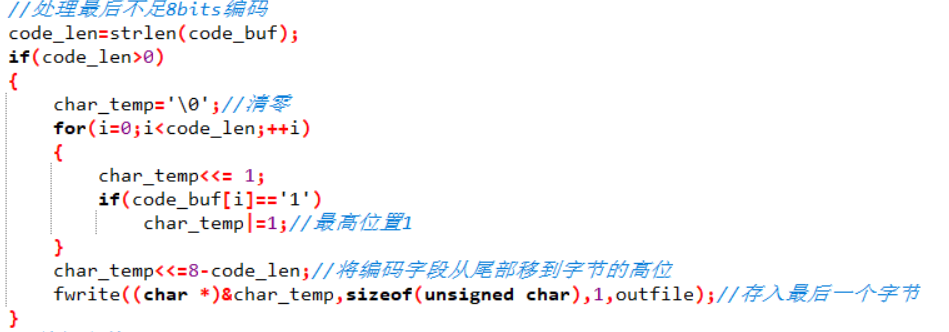
结点包含了无字符的结点以及每类字符出现的频度。而哈弗曼树结点则包含了该结点，以及字符对应的编码和双亲的左右孩子。

（2）压缩时写入字符和相应权重



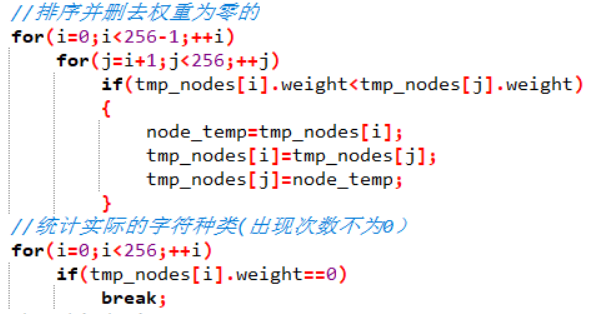
为了保证解压缩时可以顺利的进行，则写入字符和相应的权重给解压缩重构哈弗曼树使用

（3）压缩时对不足8bits位的编码的操作



将编码字段从尾部移到字节的高位。

（4）压缩时应当先进行统计，然后进行排序并删除权重为0的。



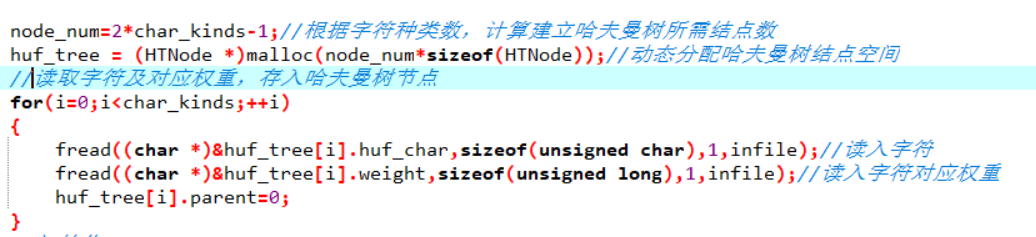
（5）.压缩时根据文件中所含字符多少进行了分类。若字符类型为1时，直接进行操作即可。若字符类型不为1时则调用创建哈夫曼树，以及哈夫曼编码的函数，以及利用哈夫曼压缩的思想来进行压缩。



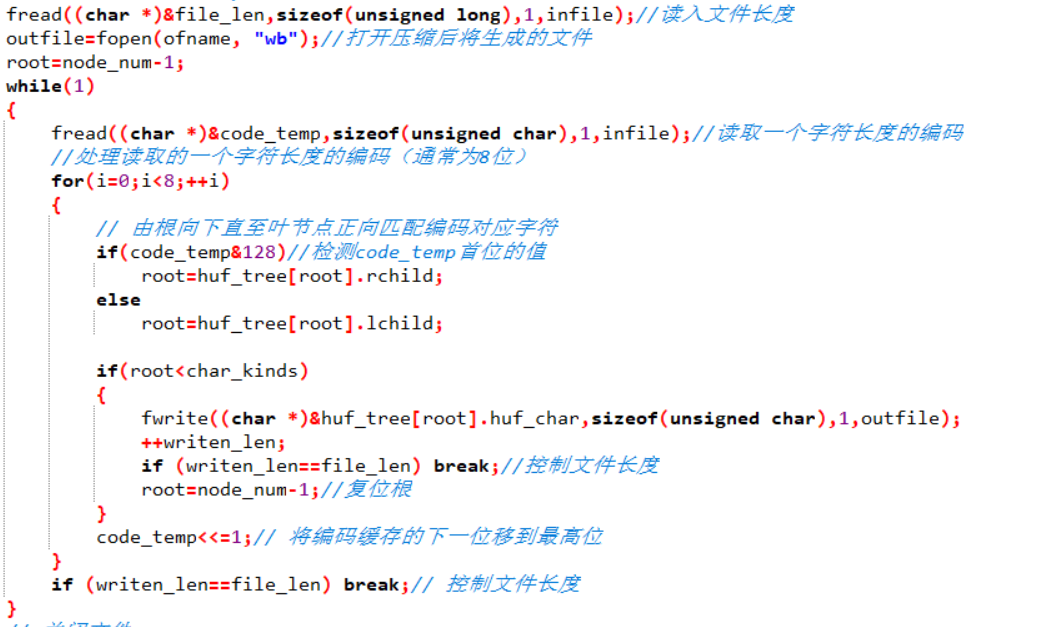
（6）解压缩时读取文件中的字符和编码，来进行哈夫曼树的重构



此部分也分为了判断字符类型是否为1.下面直接给出字符个数不为1时的情况：



此部分为读取部分，接下来是进行解码

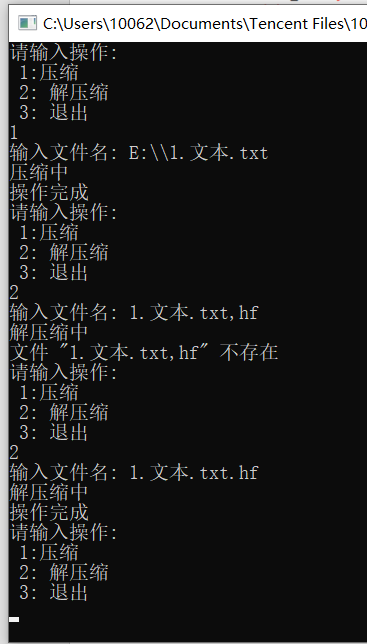


由根向下直到叶结点正向匹配编码对应的字符。

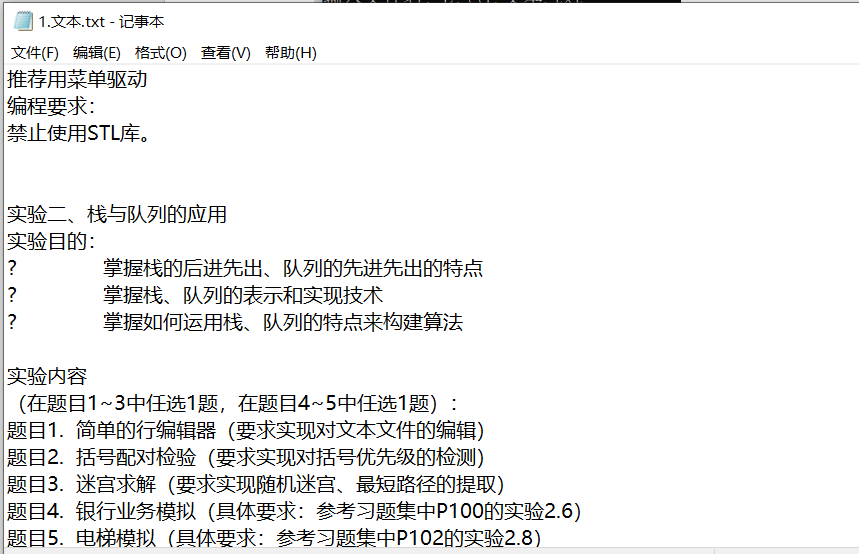
其余代码见附件！！！

（四）实验结果及其分析：

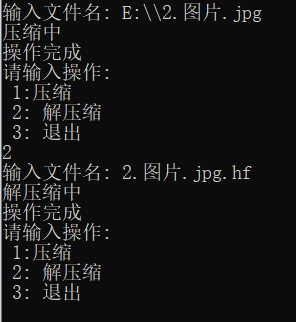
压缩和解压缩文本：







压缩和解压缩图片：







（五）debug和实验分析

1.对于初始化。

2.对于压缩时处理不足8bits的编码的处理，应该进行移位操作

3.对于文件的应用还略显不熟练，关于文件的函数应该牢记

4.对于分类，当字符类型为1时，实际上是可以不需要创建哈夫曼就可以直接写入。

5.对于解压缩时，重构哈夫曼树的过程的理解显得不太清楚。