实习报告 2 利用哈夫曼编码压缩文件

PB19030861 王湘峰

题目：利用 Huffman 树的知识设计一个压缩文件的程序

一、需求分析

对于一个特定的文件例如 example.txt，通过程序进行压缩，生成一个 example.huff 文件。之后能够用程序解压缩这个文件生成原始的 example.txt。

细节：通过命令行决定执行压缩还是解压缩，如：

Huffman.exe -z example.txt，将 example.txt 压缩为 example.huff

Huffman.exe -u example.huff 将 example.huff 解压缩为 example.txt

如果传递了错误的参数则提示错误信息并退出。

二、概要设计

为了实现该功能，需要一些函数来实现：

ADT：{

Frequency(FILE*in,long frequency[])

执行操作：统计文件中不同字符出现的频率

Create_HFtree(long weight[],HTnode[])

执行操作：构建一颗哈夫曼树

Compress（char*source_filename,char*new_filename）

执行操作：对源文件构建一颗哈夫曼树并将生成的编码写入到压缩文件中

Char_to_Bits（unsigned char [8]）

执行操作：将字符形式的哈夫曼编码转变为比特的形式

Decompress(char*source_file,char*new_file)

执行操作：从压缩文件中读取哈夫曼编码并解压缩文件

}

三、详细设计

在数据通信中，需要将传送的文字转换成二进制的字符串，用 0，1 码的不同排列来表示字符。例如，需传送的报文为“AFTER DATA EAR ARE ART AREA”，这里用到的字符集为“A，E，R，T，F，D”，各字母出现的次数为{8，4，5，3，1，1}。现要求为这些字母设计编码。要区别 6 个字母，最简单的二进制编码方式是等长编码，固定采用 3 位二进制，可分别用 000、001、010、011、100、101 对“A，E，R，T，F，D”进行编码发送，当对方接收报文时再按照三位一分进行译码。显然编码的长度取决报文中不同字符的个数。若报文中可能出现 26 个不同字符，则固定编码长度为 5。然而，传送报文时总是希望总长度尽可能短。在实际应用中，各个字符的出现频度或使用次数是不相同的，如 A、B、C 的使用频率远远高于 X、Y、Z，自然会想到设计编码时，让使用频率高的用短码，使用频率低的用长码，以优化整个报文编码。为使不等长编码为前缀编码(即要求一个字符的编码不能是另一个字符编码的前缀)，可用字符集中的每个字符作为叶子结点生成一棵编码二叉树，为了获得传送报文的最短长度，可将每个字符的出现频率作为字符结点的权值赋予该结点上，显然字使用频率越小权值越小，权值越小叶子就越靠下，于是频率

小编码长，频率高编码短，这样就保证了此树的最小带权路径长度效果上就是传送报文的最短长度。因此，求传送报文的最短长度问题转化为求由字符集中的所有字符作为叶子结点，由字符出现频率作为其权值所产生的哈夫曼树的问题。利用哈夫曼树来设计二进制的前缀编码，既满足前缀编码的条件，又保证报文编码总长最短。

首先将文件中的所有字符进行统计频率，将他们视为 Huffman 树的终端节点。使用优先队列（Priority Queue）简单达成这个过程，给与权重较低的符号较高的优先级（Priority），算法如下：

1.把 n 个终端节点加入优先队列，则 n 个节点都有一个优先权 Pi，$1 \le i \le n$

2.如果队列内的节点数>1，则：

- ⑴从队列中移除两个最小的 Pi 节点，即连续做两次 remove（min（Pi）, Priority_Queue)
- ⑵产生一个新节点，此节点为（1）之移除节点之父节点，而此节点的权重值为（1）两节点之权重和
- ⑶把（2）产生之节点加入优先队列中

3.最后在优先队列里的点为树的根节点（root）

而此算法的时间复杂度（Time Complexity）为 O（$n \log n$）；因为有 n 个终端节点，所以树总共有 2n-1 个节点，使用优先队列每个循环须 O（$\log n$）

这样从根节点往下，往左编码为 1，往右编码为 0，直至终端节

点，这样每个字符的编码就做好了。

对于字符串型的一串编码，为了将他们以更小的形式储存，我们可以通过位运算将一串 char 型的 01 串转变为比特型的 01 串。

最后是解码，通过读取压缩文件中的 Huffman 表来重建 Huffman 树，然后再将二进制字符解码为原文件中的字符，写入到生成文件中。

四、调试分析

对于文档文本的压缩，在文件小于 2KB 的时候，压缩文件大于原文件，但在文件超过 100KB 的时候，压缩率有较大的提高，平均可以将文件压缩至原文件大小的 30%~40%。

对于图片文件，压缩文件一般为原文件的 50%左右。

五、测试结果

在命令提示符窗口下输入 Huffman.exe -z t.txt

生成了 t.huff 文件

接着输入 Huffman.exe -u t.huff，生成了 t.txt，且内容与原文件一致

接着输入 Huffman.exe -a t.txt

提示 INPUT ERROR,程序退出。