【实验目的】

利用哈夫曼编码(Huffman Coding)进行文件压缩与解压缩。

为了降低难度，本次实验提供了一个哈夫曼编码的算法框架，请同学们阅读框架的说明，完善此算法框架。

【哈弗曼编码】

霍夫曼编码 Huffman Coding)是一种编码方式，以哈夫曼树—即最优二叉树，带权路径长度最小的二叉树，经常应用于数据压缩。 在计算机信息处理中，“哈夫曼编码”是一种一致性编码法（又称"熵编码法"），用于数据的无损耗压缩。这一术语是指使用一张特殊的编码表将源字符（例如某文件中的一个符号）进行编码。这张编码表的特殊之处在于，它是根据每一个源字符出现的估算概率而建立起来的（出现概率高的字符使用较短的编码，反之出现概率低的则使用较长的编码，这便使编码之后的字符串的平均期望长度降低，从而达到无损压缩数据的目的）。这种方法是由 David.A.Huffman 发展起来的。

例如，在英文中，e 的出现概率很高，而 z 的出现概率则最低。当利用哈夫曼编码对一篇英文进行压缩时，e 极有可能用一个位(bit)来表示，而 z 则可能花去 25 个位（不是 26）。用普通的表示方法时，每个英文字母均占用一个字节（byte），即 8 个位。二者相比，e 使用了一般编码的 1/8 的长度，z 则使用了 3 倍多。倘若我们能实现对于英文中各个字母出现概率的较准确的估算，就可以大幅度提高无损压缩的比例。

【哈弗曼树】

哈夫曼树又称最优二叉树，是一种带权路径长度最短的二叉树。所谓树的带权路径长度，就是树中所有的叶结点的权值乘上其到根结点的路径长度（若根结点为 0 层，叶结点到根结点的路径长度为叶结点的层数）。树的带权路径长度记为 WPL=(W1*L1+W2*L2+W3*L3+...+Wn*Ln)，N 个权值 Wi(i=1,2,...n)构成一棵有 N 个叶结点的二叉树，相应的叶结点的路径长度为 Li(i=1,2,...n)。可以证明哈夫曼树的 WPL 是最小的。

哈夫曼在上世纪五十年代初就提出这种编码时，根据字符出现的概率来构造平均长度最短的编码。它是一种变长的编码。在编码中，若各码字长度严格按照码字所对应符号出现概率的大小的逆序排列，则编码的平均长度是最小的。（注：码字即为符号经哈夫曼编码后得到的编码，其长度是因符号出现的概率而不同，所以说哈夫曼编码是变长的编码。）

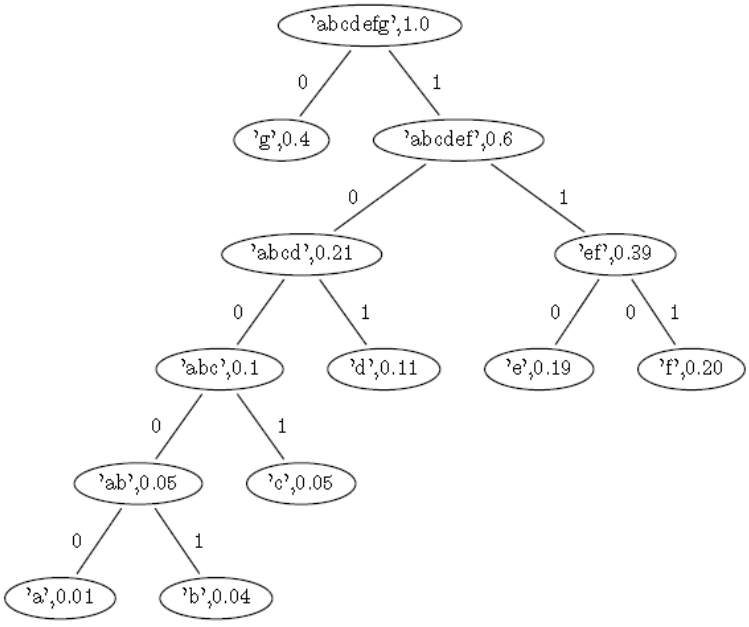

Figure 1: Huffman code tree corresponding to the given symbol frequencies

【构造哈弗曼树】

然而怎样构造一棵哈夫曼树呢？最具有一般规律的构造方法就是哈夫曼算法。一般的数据结构的书中都可以找到其描述：

一、对给定的 n 个权值{W1,W2,W3,...,Wi,...,Wn}构成 n 棵二叉树的初始集合 F={T1,T2,T3,...,Ti,...,Tn}，其中每棵二叉树 Ti 中只有一个权值为 Wi 的根结点，它的左右子树均为空。（为方便在计算机上实现算法，一般还要求以 Ti 的权值 Wi 的升序排列。）

二、在 F 中选取两棵根结点权值最小的树作为新构造的二叉树的左右子树，新二叉树的根结点的权值为其左右子树的根结点的权值之和。

三、从 F 中删除这两棵树，并把这棵新的二叉树同样以升序排列加入到集合 F 中。

四、重复二和三两步，直到集合 F 中只有一棵二叉树为止。

具体实现为：首先已知所有出现的 n 个字母的出现频率（用百分比表示）,放入一个队列中，每次从这个队列中取出现频率最小的两个字母，合并这两个字母为一个（新节点的频率为两个节点频率的和，该节点不是叶子节点，需要与叶子节点的标记区分开）作为这两个字母的父亲节点，并把该父亲节点再放入队列中。这样重复操作 n-1 次就可以得道一个 Huffman 树。最终得道的父亲节点就是该树的根节点。

【编码】

哈弗曼树是一颗二叉树，他的每个叶子节点代表一个英文字母。其上每个节点有三个指针，父亲指针（用于从叶子节点能够寻找到根节点），左右儿子节点。从根节点出发，如果我们认为左儿子为 0 右儿子为 1，这样遍历到叶子节点的时候我们就得到一个唯一的 0,1 序列了，这个序列即为这个叶子节点的字母对应的哈弗曼编码了。

【程序功能实现】

压缩：压缩的时候我们把原文章的每个字母都转换为对应的哈弗曼编码，并且把这棵树也保存起来。

解压缩：解压的时候先读取出这棵树，然后再根据这棵树把每个字母读出来。

【扩展】

读到这里，不知道同学们有没有发现一个问题。我们把每个字母都转换成了一个 0,1 串，一个字符，变成了多个字符，这样肯定变长了啊？怎么会压缩变短呢？其实，由字符生成的 0、1 序列，每位应当分别存在内存中的每个 bit 内，而非一个 char 类型内（实验所提供的框架并未实现这个功能，有兴趣的同学可以思考并实现真正的数据压缩）。

另外，有些同学可能注意到这样只能处理英文文章，中文的呢？有兴趣的同学可以上网了解中文编码规则，并尝试实现可以处理中文文章的哈弗曼编码。