文件压缩介绍：为什么要使用文件压缩呢？

在我们的现实生活中，很多的文档都存在相当的重复部分。比如一篇英文文章，即使很长但是也是由26个英文字符组成，我们只需要对只写字符进行编码，这将大大减少数据的存储空间。普遍使用的是霍夫曼编码方法。充分利用了信源该路分布的特性进行编码，它是一种最佳的逐个编号的编码方法。

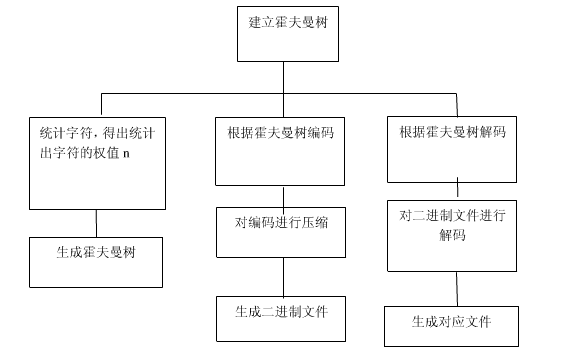
霍夫曼编码不步凑

1. 把信源符号按照其出现的概率的大小顺序排列起来
2. 把最末俩个具有最小概率的元素的概率加起来
3. 把概率之和同其余概率由大道小排序，然后再把俩个最小概率加起来，再排队
4. 重复第2,3步凑，直到概率和达到1为止
5. 在每次合并消息时，将被合并的消息赋以1和0或0和1
6. 在寻找每个信源符号到概率1处的路径，几率下路径上的1和0
7. 对每个符号写出“1”“0”序列
8. 创建霍夫曼表
9. 压缩编码时，将码值用码字代替
10. 在解码时，将码字用码值代替

基于霍夫曼编码的文件压缩与压缩的实现

1. 程序的设计思想

先是读取文件，统计文件中不同字符出现的次数，并存入数组中，然后将数组中不同的字符对应的ASCLL值存入另外数组中，按照数组各ASCLL及相应出现的频率来建立霍夫曼编码，然后将码表存入压缩文件，再将个字符对应的霍夫曼编码写入压缩文件中，实现文件压缩。解压缩时也是先把文件读入，通过霍夫曼编码的长短，依次解码，从原来的位存储还原到字节存储。



编码设计：

第一步为输入压缩软件的名称，由键盘输入文件路径和文件名称，读入字符数组中，打开该文件，按照提示进行压缩。若打不开，则继续输入。文件将信息放在字符数组中，计算每个字符出现的次数，申请一个结构体数组空间，用读取的字符减去字符结束符作为下标记录字符的概率。将所记录的字符的频率作为权值来创建霍夫曼树，依次选择权值最小的俩个字符作为左右孩子，其和作为父结点的权值，其次进行下去，知道所有的字符结点都成为叶子结点。根据创建的霍夫曼树来确定字符的01编码，左侧为0，右侧为1.读取文件，依次将每个字符用他们的编码表示，完成依次编码。

译码设计

读取编码文件，依次创建的霍夫曼树，定义一个指针指向根结点。从根结点开始，读取每一个字符，指针变化依次，当读取的字符时“1”时，指针指向当前所指结点的右侧结点，当读取的字符时“0”时，指针指向大年所指向结点的做些结点，直到指针所所指向结点为叶子结点时结束。将当前叶子结点所代表的字符值输出到姨妈文件中，依次读取编码文件中的字符，按照上述方法依次进行下去，直到文件结束。