**创新性探索课题实践报告**

# 选题: 利用概率统计方法实现英语文本压缩

## 小组成员

|  |  |
| --- | --- |
| **姓 名** | **学 号** |
| **唐成文** | **21307236** |
| **方宇豪** | **21307160** |
| **张礼贤** | **21307185** |
| **刘俊杰** | **21307174** |

## 摘要（200字以内）

面对庞杂的英文文本，如不进行压缩，将会耗费大量的体积且不宜进行传输，因此，在学习了概率论与数理统计以及数据结构中的哈夫曼树的相关知识以后，我们小组想到了借助文本中各字符出现频率构造哈夫曼树、根据哈夫曼树实现哈夫曼编码和译码的方法对文件进行压缩和解压，从而既节省了空间也方便进行文件的传输。

## 引言

在21世纪的当下，随着信息技术发展突飞猛进，甚至出现了“信息爆炸”一词以形容当下发展情况。在这种情况下，文件所占据的空间大小也在与日俱增，虽然可以通过提升磁盘容量的方法让问题得到一定的缓解，但面对磁盘无法无限扩大以及文件传输效率因为文件过大而较低的事实，我们仍然要想到另谋方法来解决这一问题，而对文件进行压缩毫无疑问就是一个不错的方法，对占据空间庞大的文件进行适当的压缩和解压，不仅能解决存储空间过大的问题，还能够在传输过程中不损坏文件的内容并能完好地随时恢复文件原来的信息，可谓一举多得。目前该课题研究大致使用了LZ77字典算法、懒惰匹配算法和Huffman编码算法。而我们组的设计主要是运用了Huffman编码算法,一方面是因为我们在数据结构的课堂中学到了该算法，对该算法较为熟悉；另一方面是Huffman编码算法整体编写较为容易，也易于理解和使用。

## 方法介绍

**方法的主要内容：**

借助文本中各字符出现频率构造哈夫曼树、根据哈夫曼树实现哈夫曼编码和译码，压缩将文本文件的每一个字符转换为哈夫曼编码、将八个位的编码为一个字符的ascii码写入压缩文件；解压将压缩文件的字符的ascii码读出，再通过哈夫曼树译码解压出正确的文本文件内容。

**动机：**

1. 我们在数据结构的课堂中学到了该算法，对该算法较为熟悉；
2. Huffman编码算法整体编写较为容易，也易于理解和使用；
3. Huffman编码能帮我们完成压缩操作，译码可以完成解压操作。

**优势：**

Huffman编码保证概率大的符号对应于短码，概率小的符号对应于长码而且所有的短码得到充分利用；且每次缩减信源的最后两个码字总是最后一位不同，前面各位相同，这两个特点保证了哈夫曼编码一定是最佳的。虽然哈夫曼编码构造出来的码不唯一，但是其平均码长是相等的，所以不影响编码效率和数据压缩性能。

## 实验结果

在指定测试数据上测试所提出的方法效果，并与一些基本方法进行比较分析，展示其优势，也可讨论其不足或可进一步优化之处。

**实验结果：**

程序运行后，得到出现字符的频率、哈夫曼编码、压缩率、压缩和解压所用的时间，英语文本压缩后的内容存放在Compress文件、解压的内容存放在Decompress文件。文件的压缩率平均在58%左右。

**优势：**

1. 文件压缩解压的执行速度快.
2. 文件的压缩率相对可观。
3. Huffman编码算法整体编写较为简单。

**不足**：

1. 字长可能还与理论上的最优编码率还有一定差距，哈夫曼编码算法是从上而下构造树。
2. 如果对应的长码一旦产生误码，某个码字的前缀部分可能成为另一个码字而发生误差，并导致错误后传。
3. 在对文件进行多次压缩时，压缩结果的大小不是每次会减少，有一个限度。
4. 同时，哈夫曼编码的压缩方法也是可能会出现压缩结果变大的可能性。当文件中字节的种类偏多，并且字节出现的次数比较平均的情况下，压缩效率就会变差，因为统计字节出现的频次信息时，各个字节出现的次数差不多，构建出来的哈夫曼树就接近平衡二叉树，效果就会不理想。当然，也会出现压缩后文件变大的情况：当哈夫曼编码的平均长度大于8字节时，压缩文件就会变大。

## 程序说明

**程序输入，输出：**

输入：无。

输出：出现字符的频率、哈夫曼编码、压缩率、压缩和解压所用的时间，得到压缩的文件和解压的文件。

**项目文件介绍：**

Text文件存放的是要压缩的英语文本(需要保证是纯英语字符)，Compress文件是英语文本压缩后的文件，Decompress文件是解压后的文件。

**运行方法：**

1、先创建三个文件Text文件、Compress文件、Decompress文件（压缩包中已给出）。

2、将要压缩的英语文本内容写进或复制到Text文件。

3、将Text文件、Compress文件、Decompress文件的地址直接在源代码中修改。

4、运行，得到出现字符的频率、哈夫曼编码、压缩率、压缩和解压所用的时间，英语文本压缩后的内容存放在Compress文件、解压的内容存放在Decompress文件。

## 参考文献

1、**Haffman编码实现文本压缩-C语言**

<https://blog.csdn.net/m15253053181/article/details/127457700?spm=1001.2014.3001.5506>

# 2、**ASCII码26 在以字符方式读取文件是产生的问题**

<https://blog.csdn.net/weixin_34232744/article/details/92807385?spm=1001.2014.3001.5506>

# 3、**基于哈夫曼编码的文件压缩**

https://blog.csdn.net/Zyc\_cucumber/article/details/119704749

## 具体分工说明

每位组员分别用200字左右介绍自己所负责的部分，总结自己在所负责的部分中做出的贡献，以及收获与体会。

**唐成文（21307236）**：

负责部分：本人在本次实践作业中主要参与了哈夫曼树的构造、哈夫曼编码和译码函数的编写。

收获：通过这次实践作业，我不仅对哈夫曼编码有了更加深刻的认识，也学会了如何在一个团队中编写程序，设计出团队需要的函数部分，此外，我也通过这次实践接触了文件的压缩与解压，进一步扩展了自己的计算机领域的知识。

**方宇豪（21307160）：**

负责部分：负责Huffman Tree的构造与编码表的构造，接收各字符及其出现的频率后对数据进行处理，得到对应的霍夫曼树，并根据树的结构给出各个字符的编码。

收获：对文件的压缩与解压有了进一步的了解，通过Huffman coding的前缀码实现来使得需要的存储空间大大减少，将在所学到的算法真正运用到了实际生活中，有种将理论付诸于实践的真实感。对Huffman coding的实际价值有

**张礼贤（21307185）：**

负责部分：本次概率论的大作业中，我负责了文件的读入和字符的统计任务，通过一个文件指针打开文件，并将文件中的字符依次读入，用一个整数数组将每个字符的出现次数进行统计。之后根据统计的次数用一个double数组将每个字符出现的频率进行统计，之后便可以根据这个double数组进行哈夫曼树的实现。除此之外，我也将众人的函数代码进行了初步整合。

收获：通过本次的概率论实践大作业，我首先学习到了C语言关于文件的众多操作，认识到了自己的很多不足之处和短板以及需要学习的地方，同时也认识到了团队合作的重要性，在与他人合作的过程中体会到了概率论知识在实践中的应用，对此有了更深一步的体会。

**刘俊杰（21307174）：**

负责部分：本次概率论的大作业中，我负责了文件的压缩和解压:先统计每个字符出现的频率的double数组，将出现次数大于0的字符和其对应的频率放入哈夫曼的节点中，利用哈夫曼树进行编码得到每个字符的的哈夫曼编码。压缩将文本文件的每一个字符转换为哈夫曼编码、将八个位的编码为一个字符的ascii码写入压缩文件；解压将压缩文件的字符的ascii码读出，再通过哈夫曼树译码解压出正确的文本文件内容。

收获：通过本次的概率论实践大作业，我熟悉了对文件读写的操作、对文件的压缩与解压的基本原理和实现方式有了更深刻的认识，同时也认识到了团队协作的重要性，对将概率论知识应用到具体实践中有了更深的体验和实践。