**《数据结构与算法》实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验名称** | 哈夫曼编码实现文本压缩 | | | | |
| **姓名** | 陈岳阳 | **学号** | 21020007009 | **日期** | 2023/5/7 |
| **实验内容** | 1. 输入字符串，输出字符串的哈夫曼编码。 2. 输入文本文件（英文），输出哈夫曼编码文件。 3. 解压缩 | | | | |
| **实验目的** | 1. 提升代码能力。 2. 更加熟练的掌握哈夫曼编码。 3. 熟练使用文件读写，bitset操作，为操作系统的学习打好基础。 4. 了解贪心的思想和优先队列、哈希表的使用。 | | | | |
| **实验步骤** | 1. 定义struct HuffmanNode，内含参数int weight，代表该结点权值；string symbol，代表该结点及其子树的符号；指向子树的lchild和rchild。 2. 编写函数Decision，用于三个实验内容的选择。 3. 编写函数Encoder，接受参数in\_path和out\_path，将文件in\_path的内容经过哈夫曼编码后写入out\_path。首先调用SymbolCount，用于统计in\_path中各字符出现的次数并记录在map m中；然后调用InitQueue，为m中的每一对(key,value)创建一个HuffmanNode，同时创建一个元素数据类型为HuffmanNode、按weight排序的小顶堆，将这些结点push进去，顶堆始终为weight最小的结点。调用GetHuffman函数，每次合并堆顶的两个结点，直到只剩最后一个，得到一棵Huffman树。最后根据Huffman编码的定义，调用InitDict函数得到Huffman编码。最后调用函数Encoder将文件压缩写入out\_path，注意C++对回车和空格的读写，文件头包含的信息以及读取方式，按位存储以及末尾不足1字节补0的细节。 4. 创建函数Decoder，调用Decode函数读取in\_path中的内容，并根据文件头中的信息将其解压写入out\_path. 5. 创建函数Str2Huffman(string str), 整体与Encoder类似，但是SymbolCount函数改为Count，用于统计str中各字符出现的次数。 6. 测试压缩，结果如下：数据集压缩 | | | | |
| **实验步骤** | 压缩后   1. 测试解压，结果：   解压解压后  8.测试字符串编码，结果：str2Huffman | | | | |
| **实验总结** | 通过这次实验，我的代码能力得到了提升，对于哈夫曼编码以及二叉树的掌握更加熟练，对文件读写、bitset有了更深的理解，对于优先队列和哈希表的使用更加得心应手，可以将贪心思想应用到问题解决中了。但是Huffman编码的压缩率和运行速度都不太理想，亟待提高。 | | | | |