**数据结构与算法程序设计**

**文本文件压缩**

# 1 项目要求

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | 文本文件压缩 | 项目类型 | 应用软件类 |
| 项目难度 | 难 | 素材资源 | 无（../res） |
| 使用工具 | 不限 | 编译系统 | Windows、Linux |
| 硬件需求 | 无 | 程序语言 | 不限 |
| 知识点 | 结构体/类、树、图、链表、查找、函数、文件操作、键盘操作等 | | |
| 项目描述 | 设计一个文本文件压缩工具，哈夫曼编码是一种常用的数据压缩技术，对数据文件进行哈夫曼编码可大大缩短文件的传输长度，提高信道利用率及传输效率。要求采用哈夫曼编码原理，统计文本文件中字符出现的词频，以词频作为权值，对文件进行哈夫曼编码以达到压缩文件的目的，再用哈夫曼编码进行译码解压缩。 | | |
| 功能实现 | * 主菜单主界面与功能一览 * 统计待压缩的文本文件中各字符的词频，以词频为权值建立哈夫曼树，并将该哈夫曼树保存到文件HufTree.dat中。 * 根据哈夫曼树（保存在HufTree.dat中）对每个字符进行哈夫曼编码，并将字符编码保存到HufCode.txt文件中。  * 压缩：根据哈夫曼编码，将源文件进行编码得到压缩文件CodeFile.dat。  * 解压：将CodeFile.dat文件利用哈夫曼树译码解压，恢复为源文件。 * 退出查询系统 | | |
| 提交材料 | 实训报告、可运行的程序 | | |

# 2 技术方案建议

### 2.1数据结构的选择

按具体需求自选数据结构

### 2.2算法参考

1．建立huffman树的时间复杂性为O(n^2)，此时n为256，所以在该过程中时间复杂度为O(1)

2、给叶子节点分配二进制码的时间复杂性为O(n\*树高)

3、输出压缩文件过程的时间复杂性为O(n)，其中n为源文件的字符个数。

4、解压缩过程的时间复杂性为O(n)，其中n为目标文件的字符个数

# 3 审查标准

### 3.1审查要点及评审标准

|  |  |  |  |
| --- | --- | --- | --- |
| **项目审查要点** | | **分值** | **得分** |
| **基本** | 主菜单主界面与功能一览 | 10 |  |
| 建立哈夫曼树 | 10 |  |
| 保存哈夫曼到文件 | 10 |  |
| 对字符进行哈夫曼编码 | 10 |  |
| 保存字符编码到文件 | 10 |  |
| 压缩得到压缩文件 | 15 |  |
| 解压得到解压后文件 | 15 |  |
| 退出工具 | 10 |  |
| **扩展** | 删除文件 | 10 |  |
| **项目总分** |  | 100 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **分值** | **得分** |
| **项目得分** | 具体见项目审查要点，若不能运行或编译则为0分，分值为项目总分\*40% | 0-40 |  |
| **文档得分** | 观点明确，论据充足，格式规范，结构清晰完整，能较好的体现项目内容 | 8-10 |  |
| 内容充实，格式规范，结构清晰，在表达或格式上略有欠缺 | 5-8 |
| 内容完整，格式有欠缺或结构不清晰 | 3-5 |
| 内容不完整，结构不清晰，未按照文档规范书写 | 0-3 |
| **用户体验得分** | 实现项目效果并有创新，界面美观，用户体验良好 | 8-10 |  |
| 实现项目基础效果，界面美观度一般，用户体验一般 | 5-8 |
| 勉强实现项目效果，用户体验较难 | 3-5 |
| 未实现项目效果，用户无法体验 | 0-3 |
| **答辩** | 思路清晰、有理有据、对于重点解释清晰，有深度和创新 | 30-40 |  |
| 思路清晰，能正确回答问题，但缺少深度 | 20-30 |
| 问题需提示才能答出或回答浅显 | 10-20 |
| 不能正确回答问题 | 0-10 |
| **总分** | 优（90以上）良（80-90）中（70-80）及格（60-70）不及格（60以下） | |  |