2024秋季学期-数据结构 Project: 基于哈夫曼编码的压缩工具

> 助教: 朱启瑞 23210240098 林佳鹏 23210240083

背景介绍

文件压缩 在节省磁盘存储空间与减少传输时间上起着重要作用。通常而言，压缩机制可被分为有损压缩和无损压缩两类。顾名思义，有损压缩会在恢复压缩文件后导致内容的丢失，常用于媒体数据；而无损压缩可以完全恢复原始文件。我们常见的 .zip .rar 格式便属于无损压缩。

哈夫曼编码 (Huffman Coding) 是可变字长编码(VLC) 的一种。Huffman 于 1952 年提出这一编码方法，该方法完全依据字符出现概率来构造异字头的平均长度最短的码字，有时也称之为最佳编码。

在本项目中，你需要使用 **哈夫曼编码** 实现一个能够对文件与文件夹进行无损压缩并且支持加密的 **压缩/解压** 工具。共需 **提交 2 次（中期检查、最终提交）文档或代码** 。项目需使用 **Java/C++ 完成，无初始代码** 。

建议在开始前先思考“中期检查”中的问题，并做项目整体的构思与设计。

请注意：切勿抄袭！！ 代码提交后将进行查重， **一经发现抄袭或雷同双方均作 0 分处理** 。

截止日期

中期检查

2024年11月17日 23:59

提交 pdf 格式的 **中期文档** 并将 **目前已完成的源代码** 打包成 zip 格式提交到 elearning。两个提交窗口分别提交，文件名格式：文档 学号_姓名_中期文档.pdf ，代码 学号_姓名.zip 。在中期检查时你应该已经搭建好项目的代码框架，未完成的代码可以用注释代替。

最终提交

2024年12月8日 23:59

提交 pdf 格式的 **开发文档** 并将 **源代码** 打包成 zip 格式提交到 elearning。两个提交窗口分别提交，文件名格式：文档 学号_姓名_开发文档.pdf ，代码 学号_姓名.zip 。

此外，还会组织全体同学面试，同学需要在自己的电脑上对该工具进行现场演示。面试时间另行通知。

具体要求

1. 中期检查 (10%)

中期文档中，需要思考并回答以下 7 个问题：

(1) 哈夫曼编码算法是否总能保证最优压缩？如果不一定，请举例说明在哪些情况下它可能不是最优的。

(2) 如何根据文件的字节流，构建哈夫曼树？

(3) 对于一组频率已经给定的字符，如果哈夫曼树已经构建完成，是否能快速找到一个新的字符的编码，而不重新构建整棵树？如何实现？

(4) 假设对一个文件进行了哈夫曼编码压缩，如何通过编码表和压缩后的字节流，准确还原出原始文件内容?

(5) 构建哈夫曼树的过程中，如何每次高效、便捷地选出出现频率最低的两个节点?

(6) 如何完成文件夹的压缩并保留内部文件名等信息的一致性?

(7) 如果需要对大量的小文件进行压缩，而不是单个大文件，哈夫曼编码的效率如何？是否有优化的空间?

(8) 于文档中附上目前代码完成情况的主体部分，并做 **简略的说明。**

若中期文档完成度较差，或是代码进度过慢，**将视情况酌情扣分** 。

2. 核心需求 (70%)

(1) 文件的压缩与解压 (30%)

需要能够正常压缩/解压给定的 **一个** 非空文件。你需要确保文件在压缩/解压操作后的内容和原始文件是完全一致的，并确保在大多数情况下压缩后的文件大小应小于压缩前的文件大小。此外，文件可能会比较大 (size > 4GB) ，你需要小心 int 溢出。(20%)

需要能够正常压缩/解压 **一个** 空文件 (size = 0B) 。(5%)

压缩时，应当能够 **指定压缩包的名称** ； 解压时，需 **还原出原本的文件名** ，即便压缩包名称与文件名不同。(5%)

(2) 文件夹的压缩与解压 (20%)

需要能够正常压缩/解压给定的 **一个** 非空文件夹。注意文件夹的 **深度** 是不确定的，即给定的文件夹中可能还有子文件夹。例如下面的 Folder 文件夹中还有 SubFolder1 和 SubFolder2 两个子文件夹，SubFolder1 下还有一个 SubFolder3 子文件夹。(10%)

需要能够正常压缩/解压 **一个** 空文件夹。(5%)

解压时，同样也应还原出原本的文件名、文件夹名。(5%)

```
Folder
├── SubFolder1
│   ├── SubFolder3
│   │   └── file1.txt
│   ├── file2.txt
│   └── file3.txt
├── SubFolder2
│   └── file4.txt
└── file5.txt
```

(3) 设置压缩密码 (15%)

在用户压缩文件或文件夹时，需要能够由用户指定或由工具直接生成压缩密码（实现其中一个即可），密码的长度和格式不作要求。在解压缩时，工具需要判断该压缩包是否为加密压缩，如果是加密压缩，则要求用户输入密码；如果不是，则直接解压。(10%)

在加密的实现上，可以直接密码编码进压缩文件的文件头中并在解压时校验；也可以使用加密算法，例如AES，对压缩文件的文件体进行加密。以上只提供两种思路，合理即可。(5%)

(4) 代码风格 (5%)

你的程序应保持良好的面向对象风格，良好的代码风格、注释习惯，具备较强的可读性，并符合标准命

名规范。不宜出现过长的类或方法，过量的耦合，或是大篇幅的重复代码。最初写出的代码很可能需要经过大规模耐心细致的重构。此部分按完成情况酌情给分。

3. 其他需求 (20%)

(1) 用户交互 (5%)

你的 PJ 需要在控制台 **以参数的形式指定输入输出**，**不断等待用户的新的指令。**
可以参考 Linux 下 tar, zip 等工具的输入输出方式：

例如 zip png.zip 1.png 表示将当前目录下的 1.png 压缩成 png.zip 。又例如 unzip png.zip 表示解压当前目录下的 png.zip。

关于 tar 和 zip 等工具的具体用法，可参考 https://www.runoob.com/w3cnote/linux-tar-gz.html

请注意： 以上只是提供一种思路，合理即可。

(2) 鲁棒性 (5%)

用户在使用工具解压的时候可能会输入错误参数，或是解压一个不是由我们的压缩工具创建的文件（例如尝试解压一个 .mp4 文件）。

对于这种情况，工具应给出类似于 **"无法解析文件格式"** 的提示，而不是直接崩溃报错，或给出一些用户看不懂的信息。实现方式不限。

(3) 文件覆盖问题 (5%)

在压缩/解压的时候可能会遇到文件覆盖 (overwrite) 问题。

例子1：用户想将当前目录下的 data_structures.txt 压缩成 ds.huffman 。但是当前目录下ds.huffman 文件已经存在。这时是否要覆盖掉原来的文件（丢失原有信息）？ 还是停止压缩?

例子2：用户想要将 ds.huffman 解压到当前目录。 ds.huffman 中包含了文件data_structures.txt ，但当前目录下 data_structures.txt 文件已存在（可能与压缩包中的不同）。这时是否要覆盖掉原来的文件（丢失原有信息）？ 还是停止解压?

请设计一个方案，**防止用户在不知情的情况下发现自己的文件被覆盖，并且可以自由选择覆盖或是停止。**实现方式不限。

(4) 开发文档 (5%)

开发文档 (PDF 格式) 应至少包含以下内容：

代码结构概要说明；

项目"核心需求"与"其他需求"中每个评分项的设计、实现思路的大致描述；

开发环境/工具，以及如何编译/运行项目；

性能测试结果（表格记录每个测试用例的初始大小、压缩后大小、压缩率）；

遇到的问题和解决方案；

其他你想说明的问题（若有）。

评分汇总

评分项	分数
中期检查	10

评分项	分数
文件的压缩与解压	30
文件夹的压缩与解压	20
设置压缩密码	15
代码风格	5
用户交互	5
鲁棒性	5
文件覆盖问题	5
开发文档	5

测试用例

123 云盘链接: https://www.123pan.com/s/VmwWTd-T5mhH

提取码:tAiA

提示与建议

1. **使用任意你喜欢的工具开发：** 你可以使用任意语言，任意编辑器/IDE，在任意平台（Windows/macOS/Linux）上进行开发。请在说明文档中注明你使用的开发环境/工具，并说明如何编译你的 PJ 。

2. **哈夫曼树的序列化与反序列化：** 在压缩前，你需要考虑如何将内存中的哈夫曼树 **存储（序列化）** 到硬盘上；并在解压前将硬盘上存储的哈夫曼树 **恢复（反序列化）** 到内存中。

3. **注意空文件和空文件夹：** 如果设计不当，Corner Case 可能会使你的程序崩溃。

4. **使用带缓冲的输入输出**：使用不带缓冲的 IO 方式，逐字节读写文件是非常慢的。你应减少你的程序在 IO 上的时间开销。

5. **注意内存消耗**：在压缩/解压时，你不应该一次性将整个文件读取到内存中，以免内存消耗过大。

6. **十六进制编辑器**：在调试 PJ 时经常需要以二进制/十六进制查看输入/输出文件内容。

7. **检验压缩->解压后的文件是否与原始文件一致**：你应该确保你的 PJ 具有无损压缩/解压的能力。

8. **尽早动手！！** 课程 PJ 工作量较大，建议不要在 DDL 前临时赶工，否则极有可能无法完成或是存在漏洞。