Project 1 - Zip Tool with Huffman Coding

Introduction

File compression is an important part in saving disk storage and helping reducing transmission time. Generally, there are two types of compression mechanisms - lossy compression and lossless compression. As their name implies, the lossy compression, which is often used for media data, would lead to content loss after recovering the compressed file, while lossless compression can recover exactly the original file. The compression format often seen in computers such as .zip, .rar belongs to lossless compression.

In this project, a tool with compression and decompression is required to be implemented by using Huffman Coding. After your submission, a face-to-face interview is also required, so please remember to make an appointment for your interview as soon as possible after finishing.

Requirements

- In Java
- An executable program with
  - compression function using Huffman coding
  - decompression function
  - user-friendly GUI (Optional, you can also use terminal commands)
  - support both folder and single file
- A user manual
- A development document which should includes
  - your design and implementation in detail
  - performance test results such as compression rate, compress and decompress time for each provided test case
  - problems encountered in this project and your solutions
  - techniques you have used for optimizing
- Optional
  - comparison of compress time and rate between your solution and other compressing tools such as WinRAR, 7Zip, HaoZip, etc

Submission

Please put all your source codes, required documents, and an **executable** file into a zip file with file name **StudentID-Name.zip** and upload it to FTP server before **2019/11/3 23:59 (GMT+08:00)**.

Besides, a face-to-face interview is also required to grade your project, so please remember to make an appointment for your interview after finishing. The deadline for interview is **2019/11/10**.

Grading

Content	Points
Single file compression & decompression	30
Folder compression & decompression	30
Performance (Time & Rate)	15
User Manual	10
Development Documentation	10
Optimization & Comparison	5

TIPs

- Think carefully about
  - how to store the huffman tree so that it can be recovered efficiently in decompress
  - how to recover characters since the length of their encoding is not fixed
- Many file types are already compressed, like JPEG and MPEG. You can have a try doing compression with standard tools for those files to see what will happen.
- The running time is somewhat relevant to the computer hardware, so please remember to include your computer's configuration along with your efficiency test.
- Avoid concatenation or subtraction of strings can help reduce the running time.
- Be care of EOF in compression and decompression.
- Be care of empty file and empty folder.
- Make use of debugger and remember to write comments.
- If encountering any problem, try to think on your own at first and then search for solutions online.
- If you find any ambiguity in this document, please contact TA.