Project 1: File Compression

Zheran Fang

8 Oct 2014

The main purpose of this project is to review the tree structure and to get you familiar with Huffman Coding and file compression. This is an individual assignment; you may not share code with other students. Java is the acceptable programming language.

Introduction

Data compression is the process of encoding information using fewer bits than the original representation would use. The essence of data compression is to reduce the length of bit encoding. There are two types of data compression: lossless data compression (decompression reproduces the original data exactly) and lossy data compression (decompression reproduces an approximation of the original data).

The lossy compression is mainly used for media files, such as image data, movie data and audio data. For example, DVDs use the lossy MPEG-2 Video codec for video compression. On the other hand, lossless compression format includes zip, rar and 7z. After lossless compression, computers can decompress the archive file and get exactly what it was before compression.

As you have learned in your Data Structure and Algorithm Analysis course, *Huffman Coding* can reduce the length of bit encoding. Thus it is a good way to do lossless compression. You can refer to the appendix document for more information about Huffman Coding.

Specification

This project requires that you carry out the following tasks:

1. Implement single file compression and decompression using Huffman

Coding.

2. Implement folder compression and decompression (recursive and non-recursive) using Huffman Coding.
3. Build a friendly GUI for your compression/decompression tool that you have implemented.
4. An executable jar file for your implementation.
5. Two project documents.
    1. **User manual**, which tells us how to use your tools to compress and uncompress a given file or folder.
    2. **Project development document**, in which you can write your project design in detail, the problems you have encountered, and as well as your solutions or ideas. You should also make several test cases to show the compression rate and time used with the given test files, and other files of different types (text, image, audio, executable file, etc.). Discuss the underlying reasons of your results.
6. **(Optional)** Optimize your compression tool, and compare your performance (both compression rate and time) to some standard compression tools (WinZip, 7-Zip, WinRAR, etc.). Write down the techniques you have used to optimize, and the comparison results.

Grading

- Single file compression/decompression: 30%
- Folder compression/decompression: 30%
- UI design & user manual: 15%
- Performance (compression rate & time): 10%
- Project development document: 15%
- Optional optimization and comparison with standard compression technologies 10% (bonus)

Submission

Create a zip file named *YourStudentID.*zip that contains your related files, and upload your zip file to the FTP server.

After submission , we will set up a face-to-face interview one by one, so get yourself ready for it.

Deadline

28 Oct 2014 00:00 GMT+08:00