Computer Science 335 - File Structures

Programming Assignment #4

Due Date: Thursday, Oct 31, 2019 (11:55pm)
(No late assignments accepted)

100 pts

Huff and Puff

You and your partner are to write two C++ programs; one which will compress a (text or binary) file and the other to decompress the file. The programs will be called **Huff.cpp** (compression program) and **Puff.cpp** (decompression program).

Huff.cpp

This program will compress a file using the Huffman algorithm. The file to compress needs to be requested by your program. *You may assume the file exists!*

Huff.cpp should create a new file with the same filename but with the extension **huf**. For example, if you were given the file **sample.dat** as input, then the compressed version's filename would be **sample.huf**.

A **huf file structure** is organized as follows: (as illustrated thru class examples)

1) Header metadata

- Length of source filename (integer)
- Source filename (string) - it is not null terminated
- Number of huffman table entries (integer)
- Huffman table

  Each table entry is structured as follows:

  glyph (integer) - value of glyph(0 - 256), or -1 (frequency node)
  (The eof glyph should be 256.)
  left pointer (integer)
  right pointer (integer)

  NOTE: The maximum number of entries in a huffman table would be 513 (257 possible glyphs + 256 merge(frequency) nodes.)

2) Compressed data

Puff.cpp

This program will decompress a file which has been compressed by Huff.cpp. The file to decompress needs to be requested by your program. Assume it will be a huf file.

Puff must reconstruct the original file. A file comparison will be done between the original source and the decompressed version to ensure that an accurate reconstruction was done.

Command Prompt has a file comparison utility (**fc**) which will check two files for any differences**.** Consult command prompt help for more information.

Assignment Procedure

Unless given special permission, this is a two-person project. If you have a teammate, do your work as a team. **This means that no one person is to do all the work for the team!** Stay in communication with each other, work together doing pair programming, do your job.

Please turn in the two files, **huff.cpp** and **puff.cpp** using **Easel**. Modularity of code is not an issue on this assignment. Documentation is suggested but you will not be graded in this area either. However, if you want to know what you did later on, you might want to document it.

A **bonus** of **9 points** will be awarded to the team having the *lowest combined compression AND decompression time* for a large binary file. It must compress to the correct size and the decompression must be an exact reconstruction of the original file.

Benchmarking Your Programs

When timing your programs, I will place time stamp markers in the following places:

1) Immediately after you obtain an input file name from the user (PLEASE PUT THIS IN MAIN SO I CAN FIND IT QUICKLY!) - this will mark the **START**.
2) The next-to-last statement in your program, - this will mark the **END**.

The last program statement will output to the standard output the length of time it took to execute your program. This output will be in seconds.

Test Files

In the comp335 folder (on \\cs1\classes\comp335\huffman) are a few test data files and their corresponding huffman compression. You may use these as you deem necessary. Keep in mind that these compressions were programmed by building each compressed byte from right to left (i.e. - from the low order to high order bit).