

Programming Assignment VII — Huffman Code

Due Date: 2021/12/06 [NCTU] 2022/01/06 [NSYSU]

1 Description of Assignment

In 1952 Huffman invented an optimal prefix-free code which can be used for lossless data compression. Each symbol in the source file is represented by a binary string. Unlike ASCII code, in which all symbols are represented by binary strings of fixed length (7 bits), symbols in Huffman code are represented by binary strings of variable-length. To achieve better compression ratio, shorter binary strings are assigned to frequently occurred symbols.

Design and implement efficient data structure and algorithm for computing Huffman code for each symbol in the input file. Then compress the input file, print out the compression ratio, and decode the encoded file.

2 Input

Choose text files of different sizes as the input to your program.

3 Output

For each input file, do the followings.

1. Compute the frequency and then the Huffman code for each symbol in the file. The symbols must include all characters in the file, including end-of-line and end-of-file.
2. Compress the input file, and print out the compression ratio. The [compression ratio](#) in this assignment is defined as the ratio of the size of the original file and the size of the compressed file. Note that the size of a file is measured in bytes. For simplicity, the Huffman code

table can be excluded in the counting of the size of the compressed file. To verify the sizes of the original file and the compressed file, use “dir” in Windows or “ls -Fl” in Unix/Linux systems.

3. Decompress the file generated in the previous step, and write it out to another file.
4. Show the compression and decompression of your program is correct by using “cmp in-file out-file” in Windows or “diff in-file out-file” in Unix/Linux systems, where “in-file” is the original input file, and “out-file” is the output file of your decode program.

Notes

1. Turn in your report with the following items before due day:
 - (a) Description of the design of your program and the data structures used in your program.
 - (b) List of your program with comments.
 - (c) Outputs of the compilation and the executions of your program.
 - (d) In addition to the basic requirements specified in the assignment, functions or features you have implemented.
 - (e) Interesting thing you have learned from this assignment.
2. The output of the program execution should indicate the correctness of your program. In other words, a set of comprehensive (but not necessarily exhaustive) annotated test data for the problem should be provided to show that your program is indeed correct. This can be done by carefully selecting a set of test data.
3. Include *assignment number*, *your name*, *student number* and *email address* on the *first* page of your report.
4. Print or write the report on A4 papers. Bind them together in the upper left corner.