
CSCE 321/2202 Fall 2018 Dr. Amr Goneid
Bonus Assignment (5%)

Date: Thu Nov 8, Due: Mon Dec 3, 2018

*This is a Bonus (Optional) assignment. No submissions will be accepted after the due date.
The assignment is to be an individual work and will be graded as such.*

Text Compression Utility

ASCII files can be compressed by using variable length Huffman Coding. By analyzing the different probabilities of the various symbols that occur in a given file, we can build a Huffman tree to come up with a minimal, optimized binary list of code words for these symbols.

In this assignment, you will be implementing a compression utility for ASCII-encoded text. Your program should be able to both compress and decompress text files, computing the compression ratio and the efficiency of your encoding. The steps to implement this program are as follows:

1. Determine the symbols / characters used in the file, build your alphabet and probabilities
2. Build a merge tree to find the optimal coding for each symbol
3. Encode your characters with the new binary codes
4. Save the coded characters along with your code table in the output compressed file

For decompression, your program should read the code table / tree from the compressed file and use that to recover the originally compressed text file.

Optimally, your program should have three different parameters passed to it at runtime:

1. A flag indicating whether it is being used to compress or decompress the input file
2. The path of the input text file
3. The path to write the output file (compressed file in case of compression, text file in case of decompression)

When used for compression, your program should output the compression ratio (for ASCII-encoded files where each character is written in 8 bits, the compression ratio will be $\langle L \rangle / 8$) and the efficiency (η) of your encoding.

Submission Instructions

Your source code and executable binary file should be submitted on blackboard by the due date, along with a README file that describes how the command line parameters should be passed (or help instructions to be printed by the program when no parameters are passed)

Grading

The program will be tested against a sample test file to compress it, check the size of the compressed file, and then used again to decompress the file to obtain the original text file. There will be no partial grades, if your program works and is capable of producing a smaller sized compressed file, then obtain the original file, you will get the full grade, otherwise, you get no partial grades.

Academic Integrity

Students are expected to commit to the principles of academic integrity. Your source code will be uploaded to turnitin to detect plagiarism. Any plagiarism detected will result in zero grade for the assignment.