CSCE 2202-01, -02 Term Project Fall 2020
Guidelines

The term project is **NOT OPTIONAL.** The project grade represents 20% of the total course grade.

Guidelines:

- The project is preferred to be a team work.
- A group **of 1 - 4 members** is to complete the project by the deadline and report the work in a project report (format attached).
- All group members **must be from the same section**. Group members may **NOT BE MIXED** between the the sections of 2202
- Names of group members should be sent by e-mail to the course TA Mr. Ahmed Haitham: ahmed.haitham@aucegypt.edu by **Saturday November 7th midnight. No allowance to change the names of group members once they have been sent to Mr. Ahmed.**
- **Names** of group members **must be submitted** in the final project report with **no allowance to change the names from those sent before.**
- The final project report should be submitted on blackboard.
- **Final date** of submission of the complete project report is **Sunday December 6th, 2020**.
- There will be NO late policy for the submission of the project. **No submissions will be accepted after the due date.**

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 project.

CSCE 2202-01, -02 Fall 2020
Term Project

Dr. Amr Goneid ***Date: Mon Nov 2, Due: Sun Dec 6, 2020***

Text Compression Utility

ASCII files can be compressed to smaller sizes 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 project, 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 binary 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 $< L > / 8$) 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 does not work. If your program works, it should be capable of producing a smaller sized compressed file, and it should be able to decompress the file to obtain the original file. In this case, grading will be on both the program and the project report.

CSCE 2202-01, -02 Term Project Fall 2020
Suggested Project Report Format

Project Title

Group Name/s
Department of Computer Science and Engineering, AUC

Abstract:
This is just a suggested format. You may change or modify it in the way that suits the project work
……
……
Keywords:

1. Introduction
……

2. Problem Definition
……

3. Methodology
……

4. Specification of Algorithms to be used
……

5. Data Specifications
……

6. Experimental Results
……

7. Analysis and Critique
……

8. Conclusions
……

Acknowledgements
…….

References
…….
…….

Appendix: Listing of all Implementation Codes
……..
……..