CS 180 Programming Project 1

Due Date: April 28 2009

Description: For this assignment, you will write two programs. The first will compress an ASCII text file and the second will recover the original text from the compressed file.

Input: You will be writing two programs, each of which will take two command-line parameters. The first parameter is the name of the input file: this will be the file to compress or decompress. The second is the output file, which designates where the output goes.

Assignment Part I: You should write a program with the name "compress" that will compress a text file. No additional information about the file may be stored by the compressor outside of the compressed file. You must achieve a running time that is linear in the size of the text file, although dependence on the number of bits per character is acceptable. Your compression factor should be *no worse* than a direct implementation of Huffman Code, as discussed in class.

You are guaranteed for this part that the input file contains exactly one `EOF` marker, and that this marker is (appropriately) at the end of the file.

Assignment Part II: You should write a program with the name "decompress". This will take as input a file that is the output from part I. It should reproduce the original text document, as it was beforehand.

Extra Credit is available for improved compression with reasonable running time. You could also implement a second data compression algorithm, in addition to Huffman. It is strongly suggested that you have a working implementation of Huffman before you try anything beyond it. Ideas for additional data compression algorithms are available from the TA upon request.

Programming Languages: You may use your choice of C/C++, or Java for this program. Please use one language for both parts: do not implement compress in C and decompress in Java. If you have a particular desire to use a different language, you are welcome to suggest it to us separately.

Groups: You may work by yourself or with a single partner (groups of two). Projects will be graded without regard to how many students are involved in the writing, and each project member will receive the same grade.

Academic Honesty: As students taking an upper-division computer science course, we expect you to understand and adhere to UCLA's standards of academic honesty. Course staff may compare all submitted code this quarter to code submitted by other students this quarter and to code submitted in previous quarters by using an automated plagiarism detection tool. Any instances of academic dishonesty will be reported according to UCLA policy. High level discussion of concepts is acceptable, but sharing of code is not.

Submission Requirements

- Any code you submit must be an original piece of code. All ideas must come from the textbook, course staff, or project members.
- Code - your code should be reasonably neat and well organized.
- Compilation Instructions: Your code must compile on the SEAS Windows machines. Do not include any pre-compiled code with your submissions. Please include a file describing how to compile your code on any such machines. It can be as simple as "open a command prompt, CD to the right directory, and type `javac *.java`".
- Your code must compile in such a way that typing "`[java] [de]compress inFile outFile`" at the command prompt will run the program.
- You should include a file, README.txt, with the following information:
  - Names and ID numbers of everyone in your group.
  - A brief description of how you store the compressed file.
  - If you did any extra credit work, what you did and what we should do to check it.
  - The status of your implementation. If your code compiles but does not run correctly, please let us know what it *does* do correctly.

Submissions are due via CourseWeb by 11:59PM on April 28. If you have a compelling reason to submit your assignment late, please let Michael know via email so that alternate arrangements can be made.

Grading Criteria

- Correctness:
  - Does your program correctly compress a document?
  - Does your program correctly de-compress a document?
- Timing and Efficiency
- Program Code; your code must be readable and properly commented. We won't be as strict as CS 32 would be for this, but be sure that we know what your code is trying to do at various stages.
- Description from the text document of your algorithms.