CSCI/CMPE 3333 Assignment Six
Instructor: Zhixiang Chen

In this homework assignment, I would like you to implement Huffman's algorithm.

The Huffman Algorithm: Given an input text file, do the following:

1. Perform a linear scan to gather frequencies of all the letters occurred in the file. You shall not consider letters with zero frequencies. Save the frequencies in a list L of binary tree nodes. Here, each node shall contain a letter and its frequency.
2. Sort the list L according to frequencies in increasing order.
3. Remove the first two nodes N1 and N2 with the lowest frequencies, build a new node N with a hypothetical letter (a dummy) and a frequency as the sum of these of N1 and N2, and add N1 as the left child of N and N2 as the right child of N. Then, insert N into L to keep L in sorted order. Keep doing the above process until L has only one node T.
4. The node T obtained from Step 3 is the Huffman code tree. For any node in the tree, its edge pointing to its left child, if there is one, can be interpreted as 0. Similarly, its right edge pointing to its right child, if there is one, can be interpreted as 1. The binary string along the edge path from the root to a letter at a leaf node is thus the Huffman code for the letter.
5. Use Huffman codes from Step 4 to encode the input text file and output the coded file in an output file, which is the encoded file.
6. Take the encoded file obtained in Step 5, decode it using the Huffman codes from Step 4 and save the result in another output file, which is the decoded file.

Warning: For Huffman coding, say, a letter is coded with "1100101," precisely, every "1" or "0" in the code is a bit "1" or "0", not an integer "1" or "0". That is, "1100101" has exactly 7 bits, but not 4 *8* 7 = 244 bits (assuming the integer size is 4 bytes). Therefore, you have to consider bit-level operation in order to implement Huffman coding correctly.

Test Run: Test your program to first encode and then decode the following text which was copied directly from http://en.wikipedia.org/wiki/Huffman_coding:

> *In computer science and information theory, a Huffman code is an optimal prefix code found using the algorithm developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes". The process of finding and/or using such a code is called Huffman coding and is a common technique in entropy encoding, including in lossless data compression. The algorithm's output can be viewed as a variable-length code table for encoding a source symbol (such as a character in a file). Huffman's algorithm derives this table based on the estimated probability or frequency of occurrence (weight) for each possible value of the source symbol. As in other entropy encoding methods, more common symbols are generally represented using fewer bits than less common symbols. Huffman's method can be efficiently implemented, finding a code in linear time to the number of input weights if these weights are sorted. However, although optimal among*

methods encoding symbols separately, Huffman coding is not always optimal among all compression methods.

Due Date:
The due date will be given via Blackboard.

Warning:
Any submission one week after the due date will not be accepted.

How to submit your work?
Please upload your source program files and your test results to Blackboard.