CSE 373 Winter 2003

Data Structures and Algorithms

Programming Assignment #3

Due: Friday March 14th

In this programming assignment, you will implement a lossless data compression algorithm, called Huffman coding (cf. Weiss Section 10.1.2).

The assignment has 3 parts.

In Part I, you are to read a text file and record the frequencies of each individual character. For example, if the text file was:

```
He got 30 on his exam; She got 32.
```

the frequencies would be: f(a) = 1; f(e) = 3; ...; f(h) = 2; ...; f(x) = 1;
f(H) = 1; f(S) = 1;
f(0) = 1; ...; f(3) = 2;
f(sp) = 8; f(.) = 1; f(;) = 1 .
For “newline” or “carriage return” characters, please see the test program written by Tian:
http://www.cs.washington.edu/education/courses/cse373/CurrentQtr/assignments.htm
The “end-of-file” character should not be counted, i.e., in the example above the “.” is the last character. If there are two spaces in a row, each one should be counted.

At the end of this part, you should output the characters that occur in the text with their frequencies.

In Part II, you are to build the *optimal Huffman code.* First build a binary heap (minimum property) of the characters with each node in the heap having a weight equal to the frequency of the characters. Then repeatedly do the following: Select and delete the 2 nodes with smallest weights (2 deletemin operations); make them the left child and the right child of a new node with weight equal to the sum of the weights of its children, and insert this node in the heap. The algorithm terminates when there is a single tree.

Now that you have the Huffman tree, in Part III you should traverse it and assign codes to its leaves. Assign a 0 when you go left and 1 when you go right. Record the length (in bits) and output the codes for each character found in the text.

Finally, you should compute and output the compression factor CF which is:

$$CF = 8/ABR$$
$$ABR = (\Sigma_{i=1}^{M} f_i \times L_i)/N$$

where ABR is the Average Bit Rate, f_i is the frequency of the i^{th} character and L_i the length of the code for the i^{th} character, M the number of distinct characters, and N the total number of characters in the text. The factor 8 is there in CF because in ASCII each character takes 8 bits.

You are **strongly encouraged** to start with a small example, for example one with the same character frequencies as in the text book. (It does not have to be the same characters, of course.) Then you can move on to a larger text file.

You can use either Java or C++. You can use any source code given on the Web sites of the book (see CSE 373 "For more info" page). For additional information on java, the following link will be useful: http://java.sun.com/j2se/1.4.1/docs/api/

Instructions for Turn-in will be given later.