CSE 3318 Lab Assignment 3

Due October 28

Goals:

1. Understanding of Huffman code trees.

2. Understanding of the five steps for developing a dynamic programming solution.

Requirements:

1. Use C to implement *order-preserving* Huffman coding - using the dynamic programming formulation described in Notes 7.C, p. 10.

   The input is 1) a positive integer n and 2) a sequence of n `doubles` giving the probabilities for symbols in an ordered character set. To simplify output, the character set should be indexed as 0 . . . n - 1. (Note that the matrices for optimal matrix multiplication are indexed from 1 to n.)

   Your program should output 1) the cost matrix, 2) the optimal order-preserving Huffman code tree, 3) the bit code assigned to each symbol, and 4) the expected bits per symbols $\left(\sum_i length_i \bullet prob_i \right)$ based on the generated code tree and the input probabilities.

2. Submit your C program on Canvas by 5:00 pm on October 28. Comments at the beginning of the source file should include: your name, your ID number, and the command used to compile your code on Omega (5 point penalty for non-compliance).

Getting Started:

1. Be sure to understand ordinary (greedy) Huffman codes and the dynamic programming solution for the optimal matrix multiplication ordering problem first.

2. The code for filling in the cost matrix will be very similar to optimal matrix multiplication ordering.

3. Outputting the optimal order-preserving Huffman code tree is just like outputting the tree for the optimal matrix multiplication ordering.

4. Determining the bit string for each character requires navigating a path down the tree stored within the cost matrix. Going left gives a `0`, going right gives a `1`. (Recursion is not needed.)