COMP 271 Design and Analysis of Algorithms
2003 Spring Semester
Questions for Eighth Tutorial – Week 0f April 22, 2003.
All problems are from Question Bank 4 – Solutions available there

1. Recall that in the Huffman coding problem, we are given a set of n characters along with their frequencies. We are required to represent each character by a unique codeword using 0's and 1's, such that no codeword is a prefix of another. The goal is to find such codewords that achieve maximum compression.

   (a) Construct the optimal code for the four characters a, b, c, d with frequencies 14, 3, 6, 10, respectively. How many bits are needed to encode a string containing 14 a's, 3 b's, 6 c's and 10 d's using this code? How may bits would be needed if we used 2 bits for each character? Which code is better?

   (b) Let x and y be the two characters having the two lowest frequencies. Is it true that x and y are siblings (i.e., children of the same node) in *every* optimal prefix tree? Justify your answer in 2 lines.

   (c) Consider the optimal prefix tree T_1 built by Huffman's algorithm, given n characters with frequencies $f_1 \leq f_2 \leq \cdots \leq f_n$. A key idea underlying Huffman's algorithm is that this tree T_1 is related to the optimal prefix tree T_2 for a certain set of $n-1$ characters. Answer the following in 2 lines (no justification is needed).

      (i) What are the frequencies of these $n-1$ characters?

      (ii) And what is the relationship of tree T_1 to tree T_2 (i.e., how can you obtain tree T_1 from tree T_2)?

2. (CLRS–16.2-4) Professor Midas drives an automobile from Newark to Reno along Interstate 80. His car's gas tank, when full, holds enough gas to travel n miles, and his map gives the distance between gas stations on his route. The professor wishes to make as few gas stops as possible along the way. Give an efficient method by which Professor Midas can determine at which gas stations he should stop and prove that your algorithm yields an optimal solution.

3. (CLRS–16.2-5) A *unit-length closed interval* on the real line is an interval $[x, 1+x]$. Describe an $O(n)$ algorithm that, given input set $X = \{x_1, x_2, \ldots, x_n\}$, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct. You should assume that $x_1 < x_2 < \cdots < x_n$.