Challenge Problems in Algorithms and Data Structures

The purpose of these problems is to provide a challenge to the top students, Each graduate student is required to turn in one problem. The due date is the afternoon of November 27. Submit your work to Dr. Larmore, not the TA. I plan to read your submission and offer criticisms, and I may ask you to correct it and resubmit during finals week.

Undergraduates who want a challenge may also submit. But I do not want you to work on these problems if you are not doing well in the course; instead concentrate on learning the regular material.

Coin Balance Problems

Recall the proof that no comparison sorting algorithm can sort any array of length n with fewer than $\log_2 n! = \Theta(n \log n)$ comparisons. The proof is that the computation tree of the algorithm must be a binary tree and must have at least $n!$ leaves, one for each permutation of the items.

Suppose you are given a set of n coins whose weights are almost, but not quite equal. You also have a balance scale. If you place equally many coins in each of the two trays, there are three possible outcoms: the left tray would be heavier, the right tray would be heavier, or the scale would balance. If we are allowed k weighings, we can distiguish among at most 3^k initial states.

If there are 12 coins and 11 of them have equal weight, but there is one "bad" coin whose weight is slightly different, the problem is to discover which coin is bad, and whether it's too heavy or too light. Thus the set of initial states is 24. If we are allowed three weighings, we can distinguish at most $3^k = 27$ inital states. Thus, it appears that the problem is solvable. In fact, it is solvable. The solution is published on the internet, so look it up and write it down.

If there are 13 coins and 12 have equal weight and one is bad, can you solve the problem in three weighings? The number of initial states is 26, which is less than 27, so we can't use that test to prove the problem unsolvable. However, that does not imply that there is an algorithm which gives you the correct answer (which coin is bad and whether it is too heavy or too light) in three weighings. Prove that there is no such algorithm. **Hint:** I was able to write the complete proof is nine lines of text.

The Traveler's Problem

For this problem, write pseudocode, not an actual program.

A traveler must walk from A to his home at B, staying at an inn during each night. There are n inns on the road. The i^{th} inn is d_i miles from A and charges w_i to spend the night. The traveler's home at B is d_n miles from A .The traveler can walk at most d miles per day. How can he get home at minimum cost? You may assume that $d_i < d_{i+1}$. The n^{th} inn is the traveler's home and is located at B, hence d_n is the distance from A to B, and $w_n = 0$ since he doesn't have to pay to sleep at home.

For simplicity, all numbers will be positive integers. Write an algorithm to find the minimum cost.

Your algorithm should state whether there is a solution, and if so, should compute the minimum cost solution.

The time complexity of the obvious dynamic programming algorithm is $O(n^2)$. Can you do better? There is an $O(n)$–time algorithm for this problem.