CSE 417: Algorithms and Computational Complexity	Winter 2012

Homework 3

Due Wednesday, 2/1/12

Remember: submit each problem, including extra credit, on *its own page*.

Problems

1. (14 pts) Consider a natural problem regarding an ancient technology called *Compact Discs*. You are given a sequence of n songs, where the i-th song is ℓ_i minutes long. You want to place all the songs on an ordered series of CDs (e.g., CD 1, CD 2, ..., CD k), where each CD holds m minutes. Furthermore,

   - The songs must be recorded in the given order: song 1, song 2, ..., song n.
   - All songs must be included.
   - No song may be split across CDs.

   Consider the following natural greedy approach to this problem. First, find the largest set of songs that will fit on the first CD, i.e., find the largest integer j such that $\sum_{i=1}^{j} \ell_i \leq m$. Put the first $min\{j, n\}$ songs on CD 1. If $n \leq j$, you have added all of the songs and you are done. Otherwise, recurse on songs $j+1$, $j+2$, ... and CDs 2, 3, ...

   (a) (2 pts) Provide a pseudocode implementation of this algorithm. (It doesn't necessarily need to be recursive.) You should make this as efficient as possible.

   (b) (2 pts) Analyze the time complexity of your implementation.

   (c) (10 pts) Prove that this algorithm uses the smallest number of CDs possible, given that the songs must be recorded in the provided order. Follow the proof structure given in Section 4.1 for the Interval Scheduling Problem in which you show that the greedy solution "stays ahead" of any other solution according to some measure. *Hint: there is more than one measure that works, but it might help if you define the following quantities. For a positive integer q, let $songs_G(q)$ be the number of songs the greedy solution fits onto the first q CDs, and let $songs_S(q)$ be the number of songs an arbitrary solution S fits onto the first q CDs.*

   (This problem is due to Sally Goldman.)

2. (16 pts) In Section 4.1 of Kleinberg and Tardos, an algorithm is given for the Interval Partitioning Problem (page 124). In this algorithm, the intervals are sorted by their start times and then considered in that sorted order. One could come up with other ways to sort the intervals. In this problem, you are to consider two alternative ways

to sort the intervals. For each way, decide whether the algorithm would still produce an optimal partition. If yes, prove it. If no, provide a counterexample and justify your answer. Remember, the algorithm you are to consider is exactly the same as the algorithm on page 124 except with a different first line.

(a) (8 pts) `Sort the intervals by their end times, breaking ties arbitrarily`.

(b) (8 pts) `Sort the intervals in decreasing order by their number of conflicts, breaking ties arbitrarily`. (Remember, as discussed on page 118, the number of conflicts of an interval i is the number of other requests that are not compatible with i.)

3. (10 pts) Chapter 4, Exercise 2. Do both parts. For part (a) you may assume that all edge costs are unique. For part (b), if you need a refresher on the Shortest s-t Path Problem, take a look at Section 4.4.

4. (10 pts) Chapter 4, Exercise 8.

Extra Credit

1. (10 pts) You're helping a group of physicists who are studying an unusual type of high-energy particle. In a typical experiment, they track a collection of n of these particles in a closed environment. Each of the n particles follows a linear trajectory through the environment, with the i-th particle following the line segment given by the parametric equation $(a_i + b_i t, c_i + d_i t)$ for $-B \leq t \leq B$. In other words, the i-th particle follows the indicated line segment as t ranges from $-B$ to B.

   Now, the potential energy of this collection of n particles at any given point in time is determined as follows. First, you build a *complete graph* on the n particles, treating each particle as a node, and joining each pair by an edge. Then you assign a cost to the edge (i, j) equal to the square of the distance between particles i and j. (Note that this cost is a function of time since it is based on the time-changing positions of i and j.) Finally, the potential energy of the collection of particles at a time t is equal to the total cost of the edges in the minimum spanning tree of this graph at time t.

   Now, as the particles move, the edge costs, the MST, and hence the potential energy all change over time. The physicists studying this system would like to find the value of t in the interval $[-B, B]$ at which the collection's potential energy is minimum. They're currently using an ad hoc heuristic search method designed to try finding this t, but they suspect that you know how to do better. Give a polynomial time algorithm that takes the trajectories of the n particles as input and determines the value of t in the interval $[-B, B]$ at which the potential energy of the collection of particles is minimized. Try to make this algorithm as efficient as possible. Analyze the running time of this algorithm and prove its correctness.

   (This problem is due to Jon Kleinberg.)