Problem Set 8 for CS 170

Problem 1 [Which Office Today?]

Suppose that you are a professor at Berkeley and you have two offices: one in Evans Hall and one in Soda Hall. Each day you choose to work in one of your two offices. Your productivity in Evans Hall on day i is given by the value e_i and your productivity in Soda Hall on day i is given by the value s_i. If you switch between offices on day i and $i+1$, you incur a productivity loss c. Given a sequence of n days, a *plan* is a sequence of n locations (i.e., a sequence of choices of Evans or Soda). The *productivity* of a plan is the sum of the productivities of each day minus the sum of the switching losses. The plan can begin in either Evans or Soda. The problem is the following: Given the values $\{e_i\}$ and $\{s_i\}$, and given c, find a plan that maximizes your productivity.

(a) Consider the greedy algorithm that chooses Evans on day i if $e_i > s_i$ and chooses Soda otherwise. Give a problem instance on which this algorithm fails. (Say what the correct answer is and why the greedy algorithm fails).

(b) Give an example of a problem instance in which the optimal plan must change locations at least three times.

(c) Give an efficient algorithm that computes the productivity of an optimal plan.

Problem 2 [Optimal Rebooting]

You're working for a company that runs a server that is accessed by many millions of customers per day. Let's suppose that you have an estimate of the numbers of customers (in millions) that you expect to access your server over the next n days. Denote these values as $(x_1, x_2, \ldots, x_n)$. Now the server software is not very well written, and the number of customers that it can handle per day decreases with each day since the most recent reboot. Let s_i denote the number of customers that it can handle on day i after the reboot, where we assume $s_1 > s_2 > \cdots > s_n$. If you choose to reboot on a given day, you can't serve any customers on that day. The problem is the following: Given the expected loads $(x_1, x_2, \ldots, x_n)$, and given the limits $(s_1, s_2, \ldots, s_n)$, find a plan that specifies the days on which you will reboot the server so as to maximize the total number of customers that you serve.

(a) Give an example of a problem instance with the following properties:

- There is a "surplus" of customers on each day, in the sense of $x_i > s_1$ for each i.
- The optimal solution reboots the system at least twice. What is this optimal solution?

(b) Give an efficient algorithm that computes the number of customers served by the optimal plan.

Problem 3 [Triangulation]

You are given a convex polygon P on n vertices in the plane (specified by their x and y coordinates). A *triangulation* of P is a collection of $n-3$ diagonals of P such that no two diagonals intersect (except possibly at their endpoints). Notice that a triangulation splits the polygon's interior into $n-2$ disjoint triangles. The cost of a triangulation is the sum of the lengths of the diagonals in it. Give an efficient algorithm for finding a triangulation of minimum cost. (Hint: Label the vertices of P by $1, \ldots, n$, starting from an arbitrary vertex and walking clockwise. For $1 \leq i < j \leq n$, let $A(i, j)$ be the minimum cost triangulation of the polygon spanned by vertices $i, i+1, \ldots, j$. Derive a dynamic program for $A(i, j)$.)