Algorithms
Computer Science 140 & Mathematics 168
Spring 2012
Homework 3b
Due Tuesday, February 7

1. **[35 Points] RNA Folding!** This is an algorithm design and implementation problem. Please go to the homework page on the course website to see the specification and test data for this problem.

2. **[35 Points] Hurts Car Rental!** Hurts Car Rental is experimenting with a new type of environmentally-friendly vehicle. These vehicles use a modular fuel pack that allows the vehicle to travel exactly 100 miles. (The fuel is believed to consist of a mixture of liquified Spam, Mountain Dew, and Cheetos, although the technology is proprietary and this is just speculation.) When the car needs to be refueled, the fuel pack is removed from the car and is replaced by a new one. Thus, it is possible that a fuel pack will be replaced before it is completely used up. Moreover, the driver gets no credit for the remaining fuel in the fuel pack and can only purchase a new fully charged 100 mile pack.[1]

   A driver may wish to travel on a long freeway from one point to another. Since only designated service stations sell the fuel packs, the driver needs to plan carefully where to refuel. Moreover, each service station charges a different amount for a fuel pack. Some drivers also don't want to stop very often to refuel. Consequently, these vehicles have a dial on the dashboard called the α dial. By turning the dial, the driver sets a value of α between 0 and 1. By doing so, the driver stipulates that she wishes to minimize the quantity $\alpha S + (1-\alpha)C$ where S is the number of fuel stops made and C is the total cost paid for the fuel packs. Notice that when α is set to 0, the objective becomes that of minimizing the total cost paid for the fuel packs. When α is set to 1, the objective becomes that of minimizing the number of stops. When α is set somewhere between 0 and 1, the objective is a linear combination of these.

   Consider a highway represented by a line segment. Let $p_1, p_2, \ldots, p_n$ be n points on the line segment sorted from left to right where p_1 is the starting point, p_n is the destination point, and each of these points has a service station selling fuel packs. Let d_i be the distance from point p_1 to point p_i, for $1 \leq i \leq n$. Let c_i be the cost of a fuel pack at the station at point p_i.

[1] This problem is loosely inspired by a similar one that arises in the new electric vehicle system pioneered by "Better Place". Take a look at `www.betterplace.com`.

(a) Professor I. Lai of the Pasadena Institute of Technology was hired by Hurts as an algorithms design consultant. He claims that the following greedy algorithm minimizes the total cost when $\alpha = 0$: Buy a fuel pack at p_1 (we have no choice there - we need fuel to depart on the trip). Travel to the furthest point which is at most 100 miles away. Purchase a fuel pack there. Now repeat this process of traveling as far as possible on the current fuel pack before purchasing a new fuel pack. Why was Prof. Lai fired? Specifically, explain briefly why this approach does not guarantee an optimal solution. You will need to provide a counterexample comprising a small collection of points, fuel costs, and a specific value of α. Show what Lai's solution gives for that instance of the problem and then show there exists a better solution.

(b) Assume that a given value of α has been established by the driver. Give clear and easily-readable pseudo-code for a recursive algorithm called `optimize(k)` that returns the cost of an optimal solution (i.e. the minimum value of $\alpha S + (1 - \alpha)C)$) for a trip from p_1 to p_k with fuel stops permitted at any of the consecutive service stations between p_1 and p_k.

Your pseudo-code will need to use the 100 mile travel limit per fuel pack and will need to refer to the d_i values that give the distances from p_1 to p_i and the c_i values that specify the cost of a fuel pack at point p_i. Finally, if there is no possible way to get from p_1 to p_k due to the distances between the given fuel stations, the function should return the value ∞.

(c) Describe a dynamic programming algorithm for this problem. In particular:

i. Describe what the dynamic programming table looks like and its dimensions.

ii. Describe the order in which the cells in the table are filled in and the rule used to fill in each cell. Be sure to explain how to fill in the "easy" cells at the beginning as well as the rule for filling in the other cells.

iii. Finally, briefly explain how you could use your DP table to reconstruct a minimum cost solution (the set of cities at which one should stop to refuel).

(d) What is the running time and space of your dynamic programming algorithm? Explain briefly. (No need to save space or apply the Hirschberg Method here.)

3. **[35 Points] OPTIONAL BONUS PROBLEM!** The Euclidean Traveling Salesperson Problem is the following: Given n points in the plane (corresponding to cities), find a cycle (or "tour") that visits each vertex (city) exactly once and minimizes the total Euclidean distance travelled. For example, Figure 1 shows 7 points in the plane. Figure 1(a) shows the optimal traveling salesperson tour. Each vertex is visited once and only once. Thus, a salesperson using this cycle could start at her home city, visit each city exactly once, and return home at a minimum total travel distance.

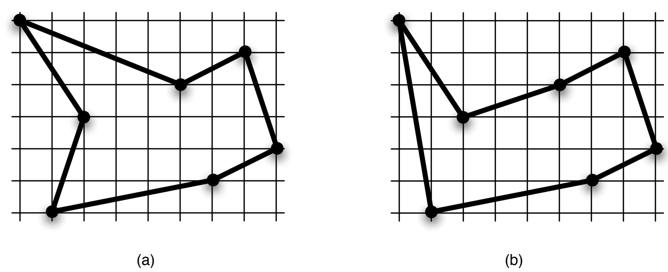

Figure 1: Seven points in the plane. (a) A shortest travelings salesperson tour. (b) A shortest "East-West" traveling salesperson tour.

Unfortunately, there is no known polynomial time algorithm for solving this problem. (Of course, we could enumerate all $O(n!)$ possible cycles and determine which one is cheapest, but that would take too long for any interesting value of n.) Worse yet, the problem is NP-complete (more about that later in the semester), meaning that if we could find an efficient algorithm for this problem then we would have instantly solved tens of thousands of open problems in mathematics and computer science. (NP-completeness is awesome!)

However, a special kind of traveling salesperson cycle called an "East-West" cycle can be found very efficiently. An "East-West" cycle starts at the westernmost point. It then heads east visiting some number of the cities until it gets to the easternmost point. It then heads back west visiting the cities that have not yet been visited. Figure 1(b) shows an optimal East-West cycle for the same 7 points shown in Figure 1(a). *You should assume that no two points have the same x-coordinate.*

Describe an efficient dynamic programming algorithm for finding an optimal East-West cycle for the Euclidean Traveling Salesperson problem in time polynomial in n. Explain briefly but convincingly why your algorithm is correct (in essence, give the sketch of the proof of correctness), and analyze the running time of your algorithm.