Algorithms Section 2
Spring 2003
Homework 4b
Due Tuesday, February 18

1. **[30 Points] Ski Optimization!** Your job at Snapple is pleasant but in the winter you've decided to become a ski bum. You've hooked up with the Triple Black Diamond Ski Resort. They'll let you ski all winter for free in exchange for helping their ski rental shop with an algorithm to assign skis to skiers.

   Ideally, each skier should obtain a pair of skis whose height matches his or her own height exactly. Unfortunately, this is generally not possible. We define the *disparity* between a skier and his or her skis to be the absolute value of the difference between the height of the skier and the pair of skis. Our objective is to find an assignment of skis to skiers that minimizes the sum of the disparities. We'll solve some interesting subproblems along the way to finding an efficient algorithm for the ski assignment problem.

   (a) First, let's assume that there are n skiers and n skis. The previous computer scientist at the ski resort, Professor R. U. Kitting, proposed the following algorithm for this case. Consider all possible assignments of the n skis to the n skiers. For each one, compute the sum of the disparities. Finally, select the assignment that minimizes the sum of the disparities. Explain why Professor Kitting was fired. How much time would this algorithm take on a computer that performs 1 billion operations per second if there were 50 skiers and 50 pairs of skis?

   (b) The next computer scientist at the resort, Dr. G. I. M. Fast, spent most of her time on the slopes. She did however make an interesting discovery one day on the chair lift. She observed that if we have a short person and a tall person, it would never be better to give the shorter person a taller pair of skis than were given to the tall person. Show that this is always true. (If you like, you may split this into cases and consider each case separately.)

   (c) After Dr. Fast's brief stint at the ski resort, you were hired. Your first job is to find a greedy algorithm that minimizes the sum of the disparities assuming that there are n people and n pairs of skis. (*Hint:* First sort the people by increasing height and sort the skis by increasing height.) Argue carefully that your algorithm does indeed find an optimal solution. In other words, show that no better solution is possible. What is the time complexity of your algorithm?

   (d) Now, your task is to design an efficient dynamic programming algorithm for the more general case that there are m skiers and n pairs of skis and $m \leq n$. Again, start by sorting the skiers and skis by increasing height. Let h_i denote the height of the i^{th} skier in sorted order and let s_j denote the height of the j^{th} pair of skis in sorted order. Let $A[i,j]$ be the optimal cost (sum of absolute differences of heights) for matching the first i skiers with skis from the set $\{1, 2, \ldots, j\}$. The solution we seek is then simply $A[m,n]$. Fill in the blanks below to define A

recursively:

$$A[i,j] = \begin{cases} \rule{7em}{0.4pt} & \text{if } i = 0 \\ \rule{7em}{0.4pt} & \text{if } 1 \le i \le j \end{cases}$$

Now give pseudocode (with explanatory comments) or a concise and complete English description of a dynamic programming algorithm that solves the optimal assignment problem of skis to skiers using a table for $A[i,j]$.

(e) What is the running time of your program? Explain.

(f) Describe how your algorithm can be modified to allow you to find an actual optimal assignment (rather than just the cost) of skis to skiers. How does this affect the running time of your algorithm?

(g) Illustrate your algorithm by explicitly filling out the $A[i,j]$ table for the following sample data:

- Ski heights: $1, 2, 5, 7, 13, 21$.
- Skier heights: $3, 4, 7, 11, 18$.

2. [**25 Points**] **Hurts Car Rental!** Hurts Car Rental is experimenting with a new type of vehicle. These vehicles use a modular fuel pack which allows the vehicle to travel exactly 100 miles. (The fuel is believed to consist of a mixture of pulverized 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.

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 $1 \le i \le n$. Let c_i be the cost of a fuel pack at the station at point p_i.

(a) Professor I. Lai 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. Explain briefly why this approach does not guarantee an optimal solution when $\alpha = 0$.

(b) Assume that a given value of α has been established by the driver. Give pseudo-code for a recursive algorithm called `optimize(`$[p_1, p_2, p_3, \ldots, p_j]$`, `$p_k$`)` which returns 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_j. You should assume that $j < k$. Moreover, we must always purchase a fuel pack at p_1 to start the trip and this counts as one fuel stop. (Notice that once you've written this function, we can call it with `optimize`$([p_1, \ldots, p_{n-1}], p_n)$ to get our desired solution!)

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 which give the distances from p_1 to p_i and the c_i values which 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 how your algorithm could be converted into a dynamic programming algorithm. 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.

(d) What is the running time of your dynamic programming algorithm? Explain very briefly.