Algorithms
Computer Science 140 & Mathematics 168
Spring 2012
Homework 3a
Due Thursday, February 2

You are no longer required to use LaTeX on your assignments, but you're certainly encouraged to use it!

1. **[30 Points] Ski Optimization!** You've landed a consulting gig at the famed Forty-Two Black Diamonds 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 between all of the skiers and their skis.*

   The input to this problem is an array of n skiers (represented just by their heights) and an array of $m \geq n$ pairs of skis (each ski pair is just the height of the skis). These arrays are given in sorted order from shortest to tallest.

   (a) Your predecessor at the resort was the infamous Prof. I. Lai from the Pasadena Institute of Technology. He proposed the following simple and fast "greedy" algorithm for the problem. The skiers and skis are already sorted from shortest to tallest. So, consider the skiers one-by-one from shortest to tallest. For each skier under consideration, assign that skier the pair of skis that most closely match that skier's height. Then remove that skier and pair of skis from consideration and repeat the process.

   Find a small example where Professor Lai's algorithm gives a solution that is worse than optimal. You'll need to provide a particular small set of skier heights, a small set of ski heights, show the solution that Professor Lai's algorithm would produce, and then show a solution that is better.

   (b) Often, we need to infer (and prove) a mathematical fact about our problem that we can exploit (exploiting a mathematical property is the one "good" kind of exploitation!) in developing our algorithm. Professor Sue

Persmart (the consultant hired after Prof. Lai was fired) made the following conjecture: If we have a short person and a tall person, it is never better to give the shorter person a taller pair of skis than were given to the tall person. Your task is to prove this! One way to do this is to break the analysis up into a number of cases. For example, let x and y be the heights of two skiers with $x < y$ and let s_x and s_y be the heights of the skis with $s_x > s_y$. We'd like to show that if the person with height x is given the skis with height s_x and the person with height y is given skis with height s_y, then switching the skis between these two skiers would not increase the disparity. One way to do this is to break up the analysis into a number of cases such as the case $x < y < s_y < s_x$, and the case $x < s_y < s_x < y$, etc. If you choose this approach, list the cases that you would need to consider and then analyze just **one** of these cases. (This is intended to save you time, because we trust that if you can do one case you could do them all!)

(c) Assume that there are n skiers and n pairs of skis. Describe (in English) a fast algorithm for assigning skis to skiers, briefly explain how the proof of correctness would work, and give the running time of your algorithm.

(d) Finally, consider the general case that $m \geq n$.

i. In simple and clear pseudo-code or English, describe a recursive algorithm for solving this problem. For now, assume that "solving" means just finding the number which is the sum of the disparities in an optimal solution (that is, the sum of the differences between the skiers and their skis in an optimal solution). Make sure to describe the base cases and the recursive call(s).

ii. Next, describe how you would implement this algorithm using dynamic programming. In particular, describe what the DP table looks like and the order in which the cells would be filled in.

iii. What is the running time and space of your algorithm? (No need to save space or use the Hirschberg Method here.)

iv. Briefly, describe how you could reconstruct an actual optimal solution matching skiers with skis using your DP table.