Math 590 Homework 6 Fall 2018

Due: **Thursday**, November 8

1. **Ski Optimization!**[1]

   You have been hired as a consultant at the famed Forty-Two Black Diamonds Ski Resort. They will 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 different between the height of the skier and the pair of skis. **Your 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 represented as just the height of the skis). These arrays are given in sorted order from shortest to tallest. For example, we might have (in inches):

   ```
   skiers = [61 67 73]
   skis = [58 62 66 68 72]
   ```

   (a) [1pt] The infamous Prof. I. Lai from the Pasadena Institute of Technology is trying to steal your job. He proposes 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 Prof. I. Lai's algorithm gives a solution that is worse than the optimal solution. You will need to provide a particular small set of skier heights, a small set of ski heights, show the solution that Prof. I. Lai's algorithm would produce, and then show a solution that is better.

   (b) [1pt] Often, we need to infer (and prove) a mathematical fact about our problem that we can exploit in developing our algorithm. Another consultant, Prof. Sue Persmart, 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 (we will call this an *inversion*). Your task is to prove this conjecture!

   Specifically: 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 skies with height s_x and the person with height y is

[1] Adapted from problem sets created by Harvey Mudd College CS Professor Ran Libeskind-Hadas.

given the skies with height s_y, then switching the skis between these two skiers would not increase the disparity (i.e., removing the inversion does not make the solution any worse).

One way to do this is to break the analysis up into a number of cases such as the case $x < y < s_y < s_x$, or the case $x < s_y < s_x < y$, etc. If you choose to use 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) [1pt] 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 (using the result from part (b)), and give the running time of your algorithm.

For part (d), you may assume that you have a function that correctly implements this algorithm and optimally assigns the n skis to the n skiers, returning the optimal disparity:

```
partC(skiers[0:n], skis[0:n])
```

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

i. [3pts] In simple and clear pseudo-code or English, describe a recursive algorithm for solving this problem. For now, assume that 'solving' means just returning the number which is the sum of the disparities in an optimal solution. Make sure to describe the base cases and the recursive call(s).

ii. [2pts] 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. A picture may help.

iii. [1pt] What is the running time of your algorithm?

iv. [1pt] Briefly, describe how you could reconstruct an actual optimal solution matching skiers with skis using your DP table.