Homework 3

Robbie McKinstry, Jack McQuown, Cyrus Ramavarapu

4 September 2016

Problem 9:

A

The algorithm whereby the skier and ski whose height difference is minimized gets assigned first and then the process is repeated is incorrect.

Consider the following counter example:

$$\text{Skiers } = \{1, 2, 3\}$$

$$\text{Skis } = \{2.5, 2.6, 3.6\}$$

This greedy algorithm will produce the following pairings:

$$\text{(Skiers, Skis) } = \{(1, 3.6), (2, 2.5), (3, 2.6)\}$$

This has an average height difference of 3.5/3.

An optimal pairing would be:

$$\text{(Skiers, Skis) } = \{(1, 2.5), (2, 2.6), (3, 3.6)\}$$

This has an average height difference of 2.7/3.

B

This greedy algorithm can be shown to be correct using an exchange argument.

Let *Alg* be the process by which the greedy algorithm operates. Assume there exists some input I such that $Alg(I)$ is incorrect.

Let $Opt(I)$ be the optimal output for input I that agrees with $Alg(I)$ for the greatest number of steps.

Since $Opt(I)$ cannot equal $Alg(I)$, there must be an earliest step of disagreement. Label this step i.

At step i, let $Alg(I)$ select (x_a, y_a) and let $Opt(I)$ select (x_i, y_i).

Since for all steps $t < i$, $Opt_t(I) = Alg_t(I)$, the values x_a and y_a must be used at some step after i in $Opt(I)$; however, x_a and y_a need not be used together in the same step.

Additionally, since $Alg(I)$ always picks the two smallest available elements and pairs them together, the pair x_a and y_a are the smallest possible values available at step i. Therefore, in $Opt(I)$, x_a and y_a are respectively smaller than any other x and y used by $Opt(I)$ after step i.

Let the step beyond i that $Opt(I)$ uses x_a and y_a be respectively u and v. Without loss of generality, let $u < v$. Define $Opt'(I)$ as $Opt(I)$ execept for all steps $k : i \leq k \leq v$ sort the x and y values in ascending order. Since x_a and y_a are the minimum values in the range $k : i \leq k \leq v$, they will be placed at step i. Hence, $Opt'(I)$ agrees with $Alg(I)$ for at least one more step than $Opt(I)$. $Opt'(I)$ is also a feasible solution since every element is still used.

To show that $Opt'(I)$ is at least as optimal as $Opt(I)$ upon sorting requires showing that $|x_i - y_i| + |x_j - y_j|$ does not increase after the sort. This follows from the following cases which fix $y_i < y_j$.

Case 1: $x_i < y_i$
In this case $x_j < x_i < y_i < y_j$. Therefore, $(x_i - y_i) < 0$ and $(x_j - y_j) < 0$. By the defintion of the absolute value, the following holds true:
$|x_i - y_i| + |x_j - y_j| = -(x_i - y_i) - (x_j - y_j) = -x_i + y_i - x_j + y_j$

This can be rearranged and the absolute values can be replaced, giving: $|x_i - y_j| + |x_j - y_i|$.
This is equivalent to the sum if the values were sorted, thereby showing that the value of the sum does not change under sorting.

Case 2: $x_i = y_i$ Since $x_i = y_i$, $(x_i - y_i) = 0$. The sum can therefore be expanded as $-(x_i - y_i) - (x_j - y_j)$ since $x_j < x_i$. Rearranging these values give and adding the absolute values gives $|x_i - y_j| + |x_j - y_i|$.

Case 3: $x_j < y_i < x_i < y_j$ Based on the inequality, the defintion of the absolute value function can be used to show $|x_i - y_i| + |x_j - y_j| = x_i - y_i - (x_j - y_j) = x_i - y_i - x_j + y_j$. Since $y_i < y_j$, $x_i - y_i - x_j + y_j > x_i + y_i - x_j - y_j$ because the positive quantity $(y_j - y_i)$ has been replaced with the negative quantity $(y_i - y_j)$. This lower bound can be rewritten as $|x_i - y_j| + |x_j - y_i|$, thereby showing that the sorting of the values actually reduces the sum.

Cases such as $y_i < x_j < x_i < y_j$ and $y_i < y_j < x_j < x_i$ have similar explanations showing the sum after a sort being equal to or less than the unsorted sum.

Since sorting will produce at least as optimal a result as $Opt(I)$, $Opt'(I)$ is at least as optimal as $Opt'(I)$, but agrees with more steps than $Opt(I)$ which is a contradiction, proving that there is no input I for which the greedy algorithm Alg is incorrect.