Time Spent: 1 hour

Collaborators and Resources:

Problem 3

For this problem we are trying to pair n groomsmen with n bridesmaids such that the smallest maximum value of delta is found. This problem is different from the previous problem 2 because of the fact that we don't care about the global maximum height difference- we are trying to create a pairing such that the local height difference between a groomsman i and his assigned bridesmaid j cannot be above a certain value. What this means is that the sum of all height differences is inconsequential. Even a single difference, say the height of i is 60 inches and the height of j is 72 inches, would discredit all other differences because the maximum value of delta would be considered to be 12. What we're going to do is simply sort the arrays for groomsmen and bridesmaids individually such that we can pair them exclusively based on their indices in the array. ...

1. We want to order array G and array B in order of increasing height
2. Let's say we use the sorting algorithm that I used in the previous problem (a variant of selection sort) in order that $g_1 < g_2 < ... < g_n$ and likewise $b_1 < b_2 < ... < b_n$.
3. Initialize an array of pairs P
4. For i in {1, 2, ..., n}
   (a) Pair g_i with b_i and place (g_i, b_i) in P
   (b) Remove those elements from their respective arrays
5. Return P

Claim 1. The algorithm runs in $O(2n^2+n) = O(n^2)$ time and correctly finds the optimal matching (in which the maximum local difference in heights is minimized).

Proof. My algorithm essentially sorts both lists of groomsmen and bridesmaids. Since there are n of each, the time taken for each is going to be n^2 which will become $2{\cdot}n^2$. Furthermore, the pairing will take an addition amount of time n because we have to go from i = 1 to i = n. This pairing is not within a nested loop and so we simply select the largest value based on exponents and disregarding coefficients, thus the algorithm has O(n^2) time. Assuming the algorithm only has one choice for each possible pairing it is greedy by definition. The reason it returns the most optimal pairing is that both arrays are (strictly) increasing which means that if we were to pair the jth bridesmaid with the i-1th groomsman (let's say he is even shorter) then we will possibly increase the maximum delta. In contrast if we paired the jth bridesmaid with the i+1th groomsman then we would be left with the ith value in G who would have to be paired at some point with an even taller bridesmaid j+n. In this case the maximum delta would also be increased out of all matchings. □