CS 4102
Basic Programming 1
Joseph Lee

Social Distancing Programming Assignment – Writeup

The restriction for this assignment was to come up with a solution that has more optimized runtime than a basic Θ(n^2). At the end, my algorithm was running at Θ(n*log(n)) runtime, which is the expected time complexity for the assignment.

I decided to use two helper methods to solve this problem with an array because array is easy to send to the method. First private method is the most important method that uses binary search to find the largest minimum distance between the guests at the tables, which gives the log(n) time complexity. The process is done by looping through the distances between the guests and creating a "mid" variable. For this problem, I set lower constrain equal to 0 and upper constrain to be max position – min position + 1 to ensure that we include all the possible distances. By doing so, this ensure that mid does not put the method into an infinite loop for some cases like when the largest minimum distance is one. The mid variable is calculated by adding the (highest + lowest) and then dividing it by two. Since we are dividing (highest + lowest) by two as we loop, this gives us logarithmic sort of exponential decrease in runtime. Because of that reason, this is more optimized than a Θ(n^2) approach. In the loop, there is a conditional statement that checks if the mid is possible distance by sending the mid value to the second private method. The second private method (validPosition) is a helper method that checks if the guests could be successfully arranged within the array consisting of n tables with the minimum distance between them depending on the mid value. If validPosition returns true, it sets lowest = mid because it means that the largest minimum distance must be at least mid, potentially larger. And if validPosition returns false, it sets highest = mid because that means the mid is too high to be a possible distance. Once the validPosition returns the outcome back to the first method,then the while loop recursively repeat the process to find the new updated the lowest (minimum distance). Therefore, the first private method takes log(n) time, and second private method takes n time. Hence, the time complexity of the program is Θ(n*log(n)). The space complexity of the program is Θ(n).