CS142, Programming III **Due: 23:59, Tue 03/14/2023**

Name: Majd Kawak / NetID: mkawa025 / CodeForces ID: mkawa025 / Student ID: 862273310

A. Finding the Rank

Submission#: 196208035

In this program, I start off with allocating a priority queue p_q to store all input scores and a map $rank$ to store score and it's rank. Starting with $rank = 1$ I process all elements in my p_q, $pop\_front()$ score and assigning rank to it, incrementing rank when score rank is allocated. Finally I output all numbers query to display. Time complexity is $O(log\ n + n) = O(n)$.

B. Feeding Friendsy

Submission#: 196563597

In this program, I start by taking input for T, n, m then I allocate 2 vectors $upper$ and $lower$ to store time intervals. Following sweep-line algorithm, I iterate though all intervals pointing to both $upper$ and $lower$ checking if time i is active in both or any. Taking max points Yihan can score at time i, adding it to final result at each iteration, display final result at the end. n in the number of intervals in $upper$ or $lower$, thus, Time complexity is $O(n)$.

C. Bookshelf

Submission#: 196733761

In this program, I start by taking input for n, k then I move on to take input books and store them in an $array$. I declare a function called $binarySearch$ that would take input books $array$ and perform a binary search to find best width to fit all books into k slots. Ranging from $[min...max]$, where min is the largest book width and max is the total width of all books combined. Performing binary search on range $[min...max]$ with each time possible width to check by calling a declared function $does\_fit$, this function checks if given books can fit in k slots with possible width. Total time complexity = $O(log\ n + n) = O(n)$.

F. Broomstick

Submission#: 197456519

In this program, I start by taking input for n, dir, k then I move on to taking input points (x, y) and store them in an map list map_list. Depending on input direction, map elements gets stores with x or y to represent a line. for example $(1, 2), (1, 4), (1, -5)$ all share same x value, thus, my $map\_list[1] = 2, 4, -4$. After taking all input, I implemented a sweep-line for loop to check $2 * d$ distance away from each map_list value. Total time complexity = $O(n)$ where n is the number of $x$or y parallel lines.