Kick Start 2022 - Round E

Analysis: Students and Mentors

Test Set 1

In Test Set 1, you can simply try all possible options. For each student, check all the other students as a potential mentor, and choose the highest rated possible one. The time complexity of this solution will be $O(N^2)$, which is enough for this test set.

Test Set 2

In Test Set 2, we have $\mathbf{N} \leq 10^5$, and $O(N^2)$ will take too much time, so a smarter, more efficient solution is needed. Below are two of the possible approaches.

Binary Search

If we sort an array of student ratings, then for each R_i we can find the maximum R_j such that $R_j \leq 2 \times R_i$ using binary search. The only caveat is that an extra care is required to prevent picking yourself as the highest rated possible mentor. We can prevent this from happening as follows. Make sure the binary search chooses the *rightmost* one of the several equal ratings in an array. Then if the rating of the mentor for a student i found with binary search is equal to R_i, we take rating to the left of it as an answer, or output -1 if it is the first element of an array.

The time complexity of this solution will be $O(\mathbf{N} \log \mathbf{N})$ because of sorting and performing binary search $\mathbf{N}$ times.

Two Pointers

If we have a sorted array of student ratings, we could also use the two pointer approach to solve the problem. Let us consider students one by one, from the lowest rated ones to the highest rated ones, and try to find the ratings of the mentors for them. As we do this, we will maintain a "pointer" to the index of the highest possible rating of a mentor for the current student. Initially the pointer will be at the start of the array. Notice that as your consider the student with the higher rating and try to find the mentor for them, pointer to the highest possible mentor rating can move only to the right in the sorted array of ratings. As such, we can simply try to move the pointer to the right as far as we can for each student, and it will move no more than $\mathbf{N}$ times in total. An additional care is required to prevent picking yourself as a mentor, similarly to the binary search solution.

The time complexity of this solution will also be $O(\mathbf{N} \log \mathbf{N})$ because you need to sort the array first. As a fun variation, it is also possible to trade time for space, and use the fact that $\max(R_i) \leq 10^6$. We can then sort the array of ratings using an array of size 10^6 and a variation of counting sort, and later execute two pointer part of the solution in $O(N)$. Total time complexity of the solution will be $O(\max(N, \max(R_i)))$, or $O(N)$ as $\max(R_i) \approx 10 \times N$, although in practice it might be not much faster than $O(\mathbf{N} \log \mathbf{N})$ solution.