Coding Competitions Farewell Rounds - Round C

Analysis: Immunization Operation

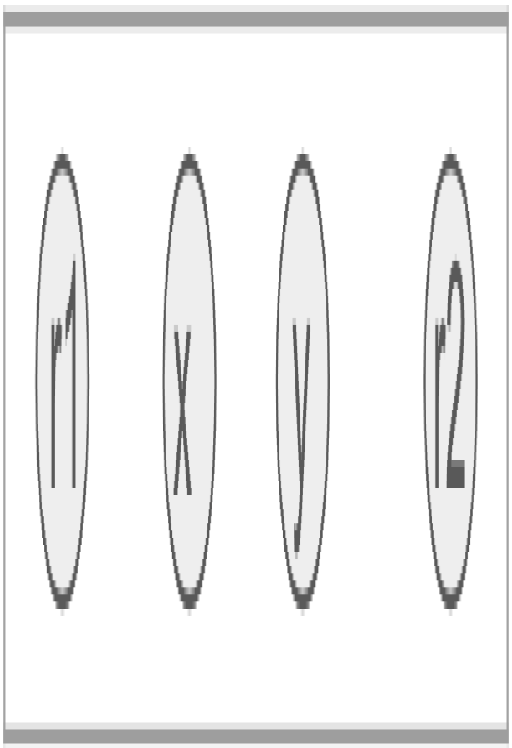
Test Set 1

We can simulate each query. First, we can store all the vaccines and the patients in an ordered data structure. Then, for each query, we can go through the range from start to end positions, maintain the list of picked up vaccines, and each time we encounter a patient whose vaccine is picked up and has not been administered yet, we mark them as vaccinated. This can give us the count of vaccinated patients for every move. For each query, there will be at most $O(\mathbf{V})$ patients and vaccines we will encounter. Saving the picked up vaccine and looking up if we can vaccinate someone with the available vaccines can be done in $O(\log \mathbf{V})$ using a set-like data structure. So the total time complexity of this approach is $O(\mathbf{M} \times \mathbf{V} \log \mathbf{V})$, which is sufficient for Test Set 1.

Test Set 2

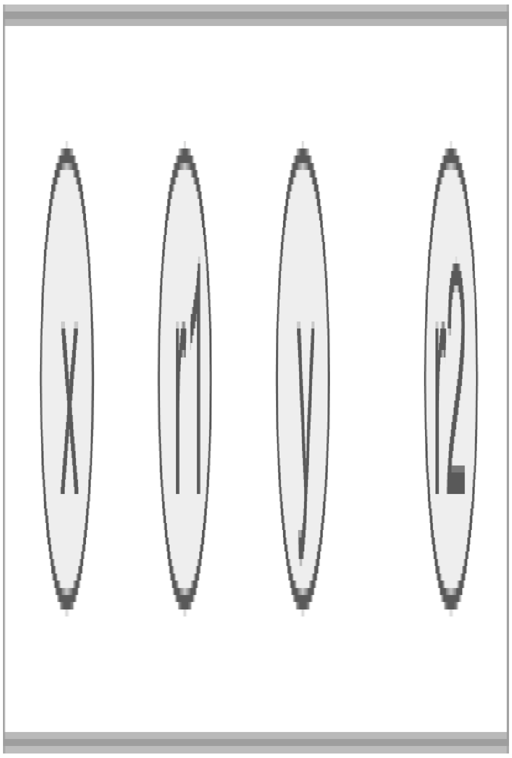
For Test Set 2, we need to optimize the simulation. One important observation here is that for each patient who is located to the east side of their vaccine, they would be vaccinated when the robot is moving in the east direction. Let us consider two cases where that can happen. The vaccine is at x, the patient at y, and y > x (the patient is on the east of the vaccine). Let us assume the move range from r_1 to r_2.

1. The move range covers both the vaccine and the patient. The vaccine will be picked up, and then the patient will be vaccinated in the same move.



2. The move covers only the vaccine, not the patient. The vaccine will be picked in this move and the robot will finish at position r_2. To complete the vaccination, the robot needs to go

to y, so it must move in the east direction to come to position y.



The same can be said about the patients who are on the west side of their vaccines, that they will be vaccinated when the robot is moving in the west direction. With this insight, we can process these two types of cases separately. Let us call them **east case** (where patients will be vaccinated in an east move) and **west case** (where patients will be vaccinated in a west move).

For the east cases, we can keep track of the farthest right point Max_R that the robot visited so far. When the robot is making an east move with the range (r_1,r_2] and r_2 is on the west of the Max_R, then this range has been already visited by the robot, and nothing new happens. Otherwise, we can count the east case patients in that range(max(r_1, Max_R), r_2], and that count will be the answer of the query. Since each patient can be vaccinated only once, this does not take longer than $O(\mathbf{V})$ time in total.

For the west cases, this simple strategy does not work, because we need to know not just the farthest right point, but also all the west case vaccines that have been picked in previous moves, and not used yet. Since the robot starts at position 0, and the vaccines and patients are always on the positive coordinates, for any west move that visits a patient, their vaccine is already picked up in one of the past east moves. So we can maintain an ordered list of west case patients whose vaccines have been picked during the previous east moves, and in each west move, find the number of patients in that move range from the list. We should later delete those patients. Since each patient will be counted and deleted only once, and we can complete each find, insert, delete operations in $O(\log \mathbf{V})$ time, the total time complexity here is $O(\mathbf{V} \log \mathbf{V})$. The overall time complexity of the whole solution is $O(\mathbf{M} + \mathbf{V} + \mathbf{V} \log \mathbf{V})$, which is sufficient for Test Set 2. $O(\mathbf{M})$ is included since we need to go through each query once at least.