Review of Werewolf

Problem

Given a connected undirected graph with N vertices and M edges. The vertices are numbered from 0 through $N-1$.

Q queries are given. The query i $(0 \leq i \leq Q-1)$ is represented by four integers S_i, E_i, L_i, R_i satisfying $L_i \leq S_i$ and $E_i \leq R_i$. You want to travel from the vertex S_i to the vertex E_i. Your route must satisfy the following condition:

- Assume that you visit the vertices $V_0, V_1, V_2, \ldots, V_p$ in this order $(V_0 = S_i, V_p = E_i)$. Then there is an index q $(0 \leq q \leq p)$ such that $L_i \leq V_0, V_1, \ldots, V_q$ and $V_q, V_{q+1}, \ldots, V_p \leq R_i$ are satisfied.

You start the travel in **human form**, transform yourself from **human form** to **wolf form** at the vertex V_q, and finish the travel in **wolf form**.

Your task is to determine whether it is possible to travel from the vertex S_i to the vertex E_i.

Subtasks and Solutions

Subtask 1 (7 points)

$N \leq 100$, $M \leq 200$, $Q \leq 100$

You choose a V vertex where you transform yourself from human form to wolf form.

For each choice of V, you need to decide whether it is possible to travel from S_i to V in human form (i.e. only using vertices whose indices are $\geq L_i$), and to decide whether it is possible to travel from V to E_i in wolf form (i.e. only using vertices whose indices are $\leq R_i$).

The time complexity of this solution is $O(QN(N+M))$.

Subtask 2 (8 points)

$N \leq 3\,000$, $M \leq 6\,000$, $Q \leq 3\,000$

Determine the set of vertices you can visit from S_i in human form, and determine the set of vertices you can visit from E_i in wolf form.

Then check whether these two sets intersect.

The time complexity of this solution is O(Q(N+M)).

Subtask 3 (34 points)

The cities are located on a line. In other words, $M = N - 1$ and no city is directly connected to more than 2 cities.

Let U_i be the set of the vertices which are reachable from S_i by passing only vertices with index at least L_i. Similarly, let V_i be the set of the vertices which are reachable from E_i by passing only vertices with index at most R_i. Note that U_i forms a range on the line on which cities are located. This range can be efficiently computed using doubling or Segment tree. V_i can be similarly computed. Then, we can answer the query by checking whether these two ranges interesect.

Subtask 4 (51 points)

No additional constraints

We can construct a rooted tree so that U_i forms a subtree. This can be done using adding vertices to a disjoint set union structure in the descending order of indices. Then, using Euler-Tour on this tree, we can obtain a sequence of vertices and every U_i corresponds to a contiguous segment of this sequence. We can compute similar sequence for V_i. Then, we can answer the query by checking whether two segments for U_i and V_i shares a vertex. This can be done by the sweep line algorithm with a segment tree. The time complexity of this solution is $O((Q+M)\log N)$.