The tiny robot

After Wolfgang has participated in the olympiad in informatics, he needed a new challange. In the Autonomously Operated Instruments (AOI) contest the most elaborated robots will be put to the test in a parcour. The goal is to develop an intelligent algorithm to steer a robot through a $n \times m$ grid (rows, columns) while avoiding obstacles. The robot starts in the bottom-left corner $(1, 1)$ and its goal is to reach the top-right corner (n, m). There are r *rectangular* obstacles which block different cells of the grid.

Immediatly Wolgang notices, that obviously it is optimal to only move the robot to the right and up, when you need to reach the right-top corner. Why should you perform a step in the opposit direction? There is only one direction for Wolfgang, forward to the goal! Since he has no clue of robotics, he cannot install eyes into his robot. Therefore, he had no other choice than to randomly move the robot to the right or to the top in each step. Worried about his robot, it should not move more than k consecuitive steps in the same direction, so the gears are not worn out one-sided.

Soon Wolfgang notices his robot stuck in the corner blocked by obstacles on the top and to the right of his small companion. Seemingly not all paths lead to the goal when only moving to the right and to the top. Also, the gear protection is unhelpful. The robot can get stuck for two reasons:

1. The cells to the right and to the bottom of the current position are marked by obstacles.
2. The robot already moved k *consecutive* steps to the right or to the top and can not move to the top or to the right from the current position. Therefore, he gets stuck because of the gear protection.

Additionally the robot may not exit the grid and can get stuck there.

With his adventurous construction Wolfgang tries his luck in the AOI. He asks himself if his mistake will be noticed at the contest. Therefore, he likes to calculate the number of possiblities the robot can get stuck. Two paths are different if they differ at any step. The number of paths can get very big, therefore, print the result modulus $10^9 + 7 = 1000000007$.

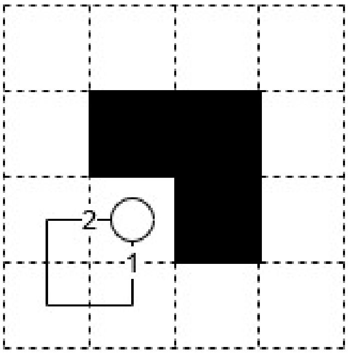
Input

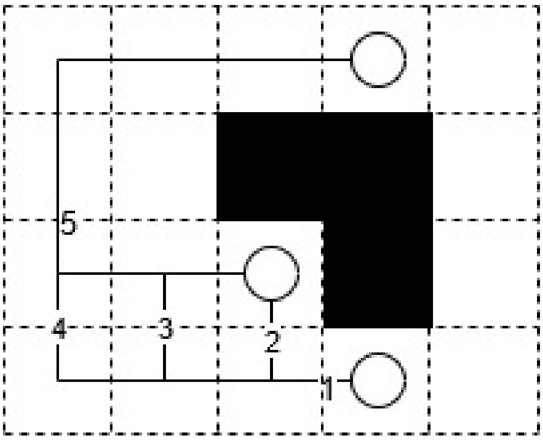
The first row contains n, m, k and r. The following r rows describe the obstacles in the format a_i, b_i, c_i, d_i. (a_i, b_i) denotes the bottom-left corner and (c_i, d_i) denotes the top-right corner of the rectangle.

Output

The number of paths the robot can get stuck.

Examples

Input	Output	Comments
4 4 4 2 3 2 3 3 2 3 3 3	2	 Two paths lead to the corner. The gear protection does not take effect.

Input	Output	Comments
4 5 3 2 3 3 3 4 2 4 3 4	5	 Path 2, 3 and 4 gets the robot stuck in the corner. In the case 1 and 5 the gear protection gets the robot stuck.

Scoring

In general:

- $1 \leq n, m \leq 2000$
- $1 \leq k \leq \max(n, m)$
- $0 \leq r \leq 10^6$
- $1 \leq a_i \leq c_i \leq n$ and $1 \leq b_i \leq d_i \leq m$ (all obstacles are contained within the grid and meaningful rectangles)
- Cell $(1, 1)$ is never blocked.

- It is possible, that there is no path to the goal or cell (n, m) is blocked.

Subtask 1 (1 Point): $r = 0$, $k = \max(n, m)$

Subtask 2 (19 Points): $k = \max(n, m)$ and exactly one rectangle with the corners $(n, 1)$ (n, m). Here the goal is not reachable and every path leads to a stop.

Subtask 3 (15 Points): $k = \max(n, m)$ and the rectangles do not overlap.

Subtask 4 (15 Points): $k = \max(n, m)$, $n, m \leq 200$

Subtask 5 (20 Points): $k = \max(n, m)$

Subtask 6 (10 Points): $n, m \leq 200$

Subtask 7 (20 Points): No additional constraints

Hint: Since the grid is sized $n \times m$, $k = \max(n, m)$ means, that the number of steps in each direction is not limited.

Constraints

Time limit: 1 s **Memory limit:** 256 MB