Problem A. Traveling in Cells 2

Little Cyan Fish (小青鱼) has a friend called Wanwan (绾绾). Wanwan is on a large grid. The coordinates of the lower left corner are $(0, 0)$, and the coordinates of the upper right corner are $(10^6, 10^6)$.

Initially, there are n obstacles on the grid, where the i-th obstacle is located in $(x_i^{(O)}, y_i^{(O)})$. All the obstacles are not on the boundary of the grid, i.e. $0 < x_i^{(O)}, y_i^{(O)} < 10^6$.

Wanwan can only move one step up or one step right at a time. At any time, Wanwan cannot occupy a cell that contains an obstacle.

Formally, a *path* from the cell (x_s, y_s) to the cell (x_t, y_t) can be represented by a series of cells $P : (a_1, b_1), (a_2, b_2), \ldots, (a_\ell, b_\ell)$, where:

- $a_1 = x_s, b_1 = y_s$ and $a_\ell = x_t, b_\ell = y_t$.
- For all $1 \le i < \ell$, exactly one of the following condition applies:
  1. $a_{i+1} = a_i$ and $b_{i+1} = b_i + 1$;
  2. $a_{i+1} = a_i + 1$ and $b_{i+1} = b_i$.
- For all $1 \le i \le \ell$, (a_i, b_i) is not an obstacle.

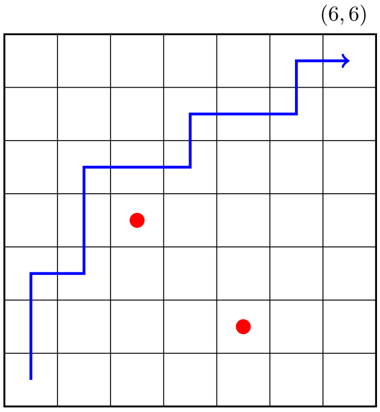

Figure 1: A valid path. Obstacles are marked as red dots.

Of course, Wanwan cannot visit any of cells with the obstacles. However, there were several more cells that was not covered by any valid path. Any cell that cannot be part of **any** valid path from $(0, 0)$ to $(10^6, 10^6)$ becomes a new obstacle. This process repeats until every free cell (i.e., a cell not containing an obstacle) can be included in some valid path P from $(0, 0)$ to $(10^6, 10^6)$.

For example, in the figure below, obstacles that appeared initially are marked as red dots, and the obstacles added in the process are marked as orange dots.

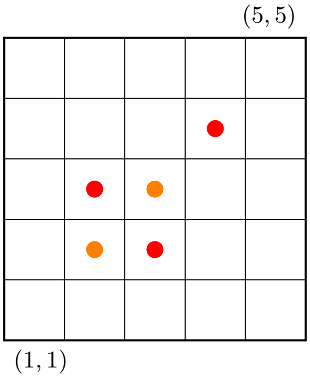

Figure 2: A grid after adding all additional obstacles.

Next, after adding all additional obstacles, Little Cyan Fish needs to complete q random walk queries. For the i-th walk query, Little Cyan Fish takes all the path from $(x_i^{(s)}, y_i^{(s)})$ to $(x_i^{(t)}, y_i^{(t)})$. Little Cyan Fish wants to know how many cells are accessible by at least one of the path.

Formally, a cell (x, y) should be counted if there exists a path $P : (a_1, b_1), \cdots, (a_\ell, b_\ell)$, such that $(a_1, b_1) = (x_i^{(s)}, y_i^{(s)})$, $(a_\ell, b_\ell) = (x_i^{(t)}, y_i^{(t)})$, and $(a_i, b_i) = (x, y)$ for some $1 \le i \le \ell$.

Input

The first line of the input contains two integers n and q $(1 \le n, q \le 10^5)$.

The next n lines of the input describe all the obstacles **existed initially**. The i-th line of these lines contains two integers $x_i^{(O)}$ and $y_i^{(O)}$ $(1 \le a_i, b_i < 10^6)$, indicating the coordinates of the i-th obstacle.

The next q lines of the input describe all the queries. The i-th line of these lines contains four integers $x_i^{(s)}$, $y_i^{(s)}$, $x_i^{(t)}$, and $y_i^{(t)}$ $(1 \le x_i^{(s)} \le x_i^{(t)} < 10^6, 1 \le y_i^{(s)} \le y_i^{(t)} < 10^6)$, indicating the i-th task.

Output

For each walk task, output a single line with a single integer, indicating the answer.

Examples

standard input	standard output
3 3 2 3 3 2 4 4 1 1 5 5 3 4 5 5 3 3 4 5	20 4 0
8 2 2 2 2 5 3 5 4 1 5 4 6 1 7 3 8 2 1 1 9 5 5 1 8 5	28 11

Note

In the first example, there are three obstacles initially (marked as red dots), and two more obstacles added after Wanwan's first trip (marked as orange dots).

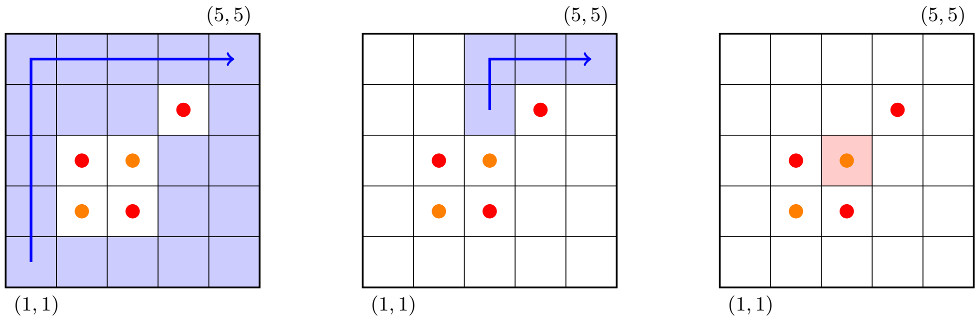

Figure 3: The figure corresponding the first example.

For each of the query, all the accessible cells are marked as blue. In the third query, as $(x_i^{(s)}, y_i^{(s)})$ has been occupied with an obstacle, there were no cells can be accessible by any of the path.