

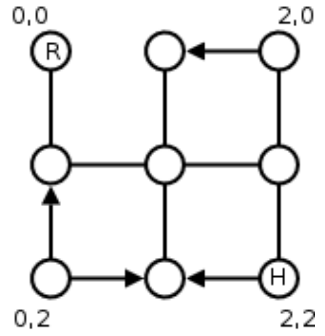
Selma and Louis are two of the most dangerous bank robbers in Man-hat'em City. The main reason they are so good, and never got caught, is their ability to create perfect escape plans. The problem is they are getting old and their mind isn't what it used to be, so they are looking for someone to create a program that automatically creates these escape routes for them.

A well known figure in the crime scene of Man-hat'em told them you were the best guy for the job.

Man-hat'em is a very well organized city. Its streets are layed out in a grid and the distance between each crossroad is always the same. However some streets are one way only and some don't even allow cars.

Besides having to make a quick getaway Selma and Louis have one other problem. The city has recently installed a surveillance system. Cameras have been installed in some crossroads but only one camera is monitored at each given time. The success of their getaway is drastically reduced if they are spotted during their escape so their escape route must avoid any crossing that is being monitored. Luckily, their friend Tony managed to get the surveillance system scheduling plans.

Your job will be to create a program that, given a certain partial map of the city and the surveillance system scheduling, will compute the optimum escape route. The following can be assumed to be always true:



- The robbery will always take place at the northwest of the map;
- The hideaway will always be at the southeast of the map;
- There is always a possible escape route;
- The time needed to get from one crossroad to the next is always the same, and, for simplicity sake, can be considered as being one time unit.
- The escape route must never pass a crossing that is being monitorized at that time;
- To avoid getting caught on camera Selma and Louis can wait any units of time at a given crossroad.

Input

The input file contains several test cases, each of them as describes below.

- The input starts by stating the number of vertical roads (nv) and horizontal roads (nh) in a single line. With $1 \leq nv, nh \leq 100$.
- The following line will contain the number (r) of traffic restrictions in the city. With $0 \leq r \leq 500$.
- Each of the following r lines will contain a restriction of the form: $x_1 y_1 x_2 y_2$. Meaning that traffic isn't allowed from the crossroad with coordinates (x_1, y_1) to the crossroad with coordinates (x_2, y_2) .
- The next line will contain the number (m) of scheduled crossroad monitorizations. With $0 \leq m \leq 500$.
- Each of the following m lines will contain the monitorization information in the form: $t x y$. Where t is the time counting from the start of the getaway and (x, y) are the coordinates of the crossroad being monitored. All these m lines will have different values for t . With $0 \leq t \leq 500$.

Output

For each test case, the output will contain a single line stating the minimum number of time units needed for a perfect getaway.

Explanation of the example output:

The plan starts at (0,0) at time 0. They then proceed to (0,1) where they intended to turn left heading to (1,1). Thanks to Tony, they know someone will be monitoring them at (1,1) by the time they get there, so they wait for one time unit. They arrive at (1,1) at time 3, where they wait another time unit as they know they will be monitored at (2,1) at time 4. After this second wait, they proceed to (2,1) and then to (2,2) where they arrive at time 6.

Sample Input

```
3 3
6
0 0 1 0
1 0 0 0
1 0 2 0
0 1 0 2
1 2 0 2
1 2 2 2
2
2 1 1
4 2 1
```

Sample Output

6