seats
Dutch (BEL)

Seats

You are going to hold an international programming contest in a rectangular hall, which has HW seats arranged in H rows and W columns. The rows are numbered from 0 through H-1 and the columns are numbered from 0 through W-1. The seat in row r and column c is denoted by (r,c). You invited HW contestants, numbered from 0 through HW-1. You also made a seating chart, which assigns the contestant i $(0 \le i \le HW-1)$ to the seat (R_i, C_i) . The chart assigns exactly one contestant to each seat.

A set of seats in the hall S is said to be **rectangular** if there are integers r_1 , r_2 , c_1 , and c_2 satisfying the following conditions:

- $0 \le r_1 \le r_2 \le H 1$.
- $0 \le c_1 \le c_2 \le W 1$.
- S is exactly the set of all seats (r,c) such that $r_1 \leq r \leq r_2$ and $c_1 \leq c \leq c_2$.

A rectangular set consisting of k seats $(1 \le k \le HW)$ is **beautiful** if the contestants whose assigned seats are in the set have numbers from 0 through k-1. The **beauty** of a seating chart is the number of beautiful rectangular sets of seats in the chart.

After preparing your seating chart, you receive several requests to swap two seats assigned to two contestants. More precisely, there are Q such requests numbered from 0 through Q-1 in chronological order. The request j ($0 \le j \le Q-1$) is to swap the seats assigned to contestants A_j and B_j . You accept each request immediately and update the chart. After each update, your goal is to compute the beauty of the current seating chart.

Implementation details

You should implement the following procedure and function:

```
give_initial_chart(int H, int W, int[] R, int[] C)
```

- H, W: the number of rows and the number of columns.
- R, C: arrays of length HW representing the initial seating chart.
- This procedure is called exactly once, and before any call to swap seats.

```
int swap seats(int a, int b)
```

- This function describes a request to swap two seats.
- a, b: contestants whose seats are to be swapped.
- ullet This function is called Q times.
- This function should return the beauty of the seating chart after the swap.

Example

Let
$$H=2$$
, $W=3$, $R=[0,1,1,0,0,1]$, $C=[0,0,1,1,2,2]$, and $Q=2$.

The grader first calls give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2]).

At first, the seating chart is as follows.

0	3	4
1	2	5

Let's say the grader calls swap_seats(0, 5). After the request 0, the seating chart is as follows.

5	3	4
1	2	0

The sets of seats corresponding to the contestants $\{0\}$, $\{0,1,2\}$, and $\{0,1,2,3,4,5\}$ are rectangular and beautiful. Thus, the beauty of this seating chart is 3, and swap_seats should return 3.

Let's say the grader calls $swap_seats(0, 5)$ again. After the request 1, the seating chart goes back to the initial state. The sets of seats corresponding to the contestants $\{0\}$, $\{0,1\}$, $\{0,1,2,3\}$, and $\{0,1,2,3,4,5\}$ are rectangular and beautiful. Hence, the beauty of this seating chart is 4, and $swap_seats$ should return 4.

The files sample-01-in.txt and sample-01-out.txt in the zipped attachment package correspond to this example. Other sample inputs/outputs are also available in the package.

Constraints

- 1 < *H*
- 1 < W
- $HW \le 1000000$
- $0 \le R_i \le H 1 \ (0 \le i \le HW 1)$
- $0 \le C_i \le W 1 \ (0 \le i \le HW 1)$
- $(R_i, C_i) \neq (R_j, C_j) \ (0 \leq i < j \leq HW 1)$
- 1 < Q < 50000
- $0 \le a \le HW 1$ for any call to swap seats
- $0 \le b \le HW 1$ for any call to swap seats
- $a \neq b$ for any call to swap_seats

Subtasks

- 1. (5 points) $HW \le 100$, $Q \le 5000$
- 2. (6 points) $HW \le 10\,000$, $Q \le 5\,000$
- 3. (20 points) H < 1000, W < 1000, Q < 5000
- 4. (6 points) $Q \leq 5\,000$, $|a-b| \leq 10\,000$ for any call to swap seats
- 5. (33 points) H = 1
- 6. (30 points) No additional constraints

Sample grader

The sample grader reads the input in the following format:

- line 1: HWQ
- line 2+i ($0 \le i \le HW-1$): R_i C_i
- line 2 + HW + j ($0 \le j \le Q 1$): $A_j B_j$

Here, A_j and B_j are parameters for the call to swap_seats for the request j.

The sample grader prints your answers in the following format:

• line 1+j ($0 \le j \le Q-1$): the return value of swap_seats for the request j