Asientos

Tú organizarás una competencia internacional de programación en una sala rectangular, la cual tiene HW asientos acomodados en H filas y W columnas. Las filas están numeradas de 0 a $H-1$ y las columnas están numeradas de 0 a $W-1$. El asiento en la fila r y columna c está representado por (r, c). Invitaste HW concursantes, numerados de 0 a $HW-1$. Además hiciste una gráfica de asientos, la cual asigna al concursante i ($0 \leq i \leq HW-1$) al asiento (R_i, C_i). La gráfica asigna exactamente un concursante a cada asiento.

Un conjunto de asientos en la sala S se dice que es **rectangular** si existen los enteros r_1, r_2, c_1, and c_2 satisfaciendo las siguientes condiciones:

- $0 \leq r_1 \leq r_2 \leq H-1$.
- $0 \leq c_1 \leq c_2 \leq W-1$.
- S es exactamente el conjunto de todos los asientos (r, c) tal que $r_1 \leq r \leq r_2$ y $c_1 \leq c \leq c_2$.

Un conjunto rectangular de asientos consiste de K asientos es **hermoso** si los concursantes a los cuales asignaron los asientos en el conjunto tienen los números de 0 a $k-1$. La **hermosura** de una gráfica de asientos es el número de conjuntos rectangulares de asientos hermosos en la gráfica.

Despues de preparar tu gráfica de asientos, recibes varias solicitudes de intercambiar dos asientos asignados a dos concursantes. Más precisamente, existen Q de tales solicitudes numerados de 0 a $Q-1$ en orden cronológico. La solicitud j ($0 \leq j \leq Q-1$) es intercambiar el asiento asignado a los concursantes A_j y B_j. Tú aceptas cada solicitud inmediatamente y actualizas la gráfica. Después de cada actualización, tu objetivo es calcular la belleza de la gráfica de asientos actual.

Detalles de implementación

Tú debes implementar el siguiente procedimiento y función:

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

- `H`, `W`: el número de filas y el número de columnas.
- `R`, `C`: arreglos de tamaño HW representando la gráfica de asientos inicial.
- Este procedimiento es llamado exactamente una vez, y antes de cada llamada a

`swap_seats`.

```
int swap_seats(int a, int b)
```

- Esta función describe una solicitu de intercambio de dos asientos.
- `a`, `b`: concursantes de los cuales sus asientos serán intercambiados.
- Esta función es llamada Q veces.
- Esta función debe retornar la belleza de cada gráfica de asientos después del intercambio.

Ejemplo

Sean $H = 2$, $W = 3$, $R = [0, 1, 1, 0, 0, 1]$, $C = [0, 0, 1, 1, 2, 2]$, y $Q = 2$.

El evaluador primero llama a `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])`.

Al principio, la gráfica de asientos está así:

0	3	4
1	2	5

Digamos que el evaluador llama `swap_seats(0, 5)`. Después de la solicitud "0", la gráfica de asientos queda de la siguiente forma:

5	3	4
1	2	0

El conjunto de asientos correspondiente a los participantes $\{0\}$, $\{0, 1, 2\}$, and $\{0, 1, 2, 3, 4, 5\}$ es rectangular y hermoso. Entonces, la hermosura de esta gráfica de asientos es 3, y `swap_seats` debería retornar 3.

Digamos que el evaluador llama a `swap_seats(0, 5)` de nuevo. Después de la solicitud 1, la gráfica de asientos retrocede al estado inicial. El conjunto de asientos correspondientes a los participantes $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$, y $\{0, 1, 2, 3, 4, 5\}$ es

rectangular y hermosa. Por lo tanto, la hermosura de de esta gráfica de asientos es 4, y `swap_seats` debe retornar 4.

Los archivos `sample-01-in.txt` y `sample-01-out.txt` en el paquete comprimido adjunto corresponde a este ejemplo. Otros ejemplos de entradas/salidas están disponibles en el paquete.

Constraints

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ for any call to `swap_seats`
- $0 \leq b \leq HW - 1$ for any call to `swap_seats`
- $a \neq b$ for any call to `swap_seats`

Subtasks

1. (5 points) $HW \leq 100$, $Q \leq 5\,000$
2. (6 points) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 points) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
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: H W Q
- line $2 + i$ ($0 \leq i \leq HW - 1$): R_i C_i
- line $2 + HW + j$ ($0 \leq j \leq Q - 1$): 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 $1 + j$ ($0 \leq j \leq Q - 1$) : the return value of `swap_seats` for the request j