



Seats

Realizarás un concurso internacional de programación en un salón rectangular, que tiene HW asientos alineados 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 se denota (r, c) . Invitaste a HW concursantes, numerados de 0 a $HW - 1$. También creaste un mapa de asientos, que asigna al concursante i ($0 \leq i \leq HW - 1$) al asiento (R_i, C_i) . El mapa asigna exactamente un concursante a cada asiento.

Un conjunto S de asientos en el salón se dice que es **rectangular** si existen enteros r_1, r_2, c_1 y c_2 que satisfacen 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 que contiene k asientos ($1 \leq k \leq HW$) es **bello** si los concursantes asignados a dicho conjunto tienen números del 0 al $k - 1$. La **belleza** de un mapa de asientos es el número de conjuntos rectangulares bellos en el mapa.

Después de preparar tu mapa de asientos, recibes muchas peticiones para intercambiar dos asientos asignados a dos concursantes. Específicamente, hay Q peticiones numeradas del 0 al $Q - 1$ en orden cronológico. La petición j ($0 \leq j \leq Q - 1$) pide intercambiar los asientos asignados a los concursantes A_j y B_j . Aceptas cada petición inmediatamente y actualizas el mapa. Después de cada actualización, tu objetivo es calcular la belleza del mapa de asientos resultante.

Detalles de implementación

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, respectivamente.
- R, C : arreglos de longitud HW representando el mapa de asientos original.
- Este procedimiento es llamado exactamente una vez, y antes de cualquier llamada a `swap_seats`.

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

- Esta función describe a una petición para intercambiar dos asientos.
- a, b : los concursantes cuyos asientos deben ser intercambiados.
- Esta función es llamada Q veces.
- Esta función debe retornar la belleza del mapa de asientos después del intercambio.

Ejemplo

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

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

Al inicio, el mapa de asientos es como el siguiente.

0	3	4
1	2	5

Decimos que el grader llama a `swap_seats(0, 5)`. Luego la solicitud 0, el mapa de asientos es como el siguiente.

5	3	4
1	2	0

El conjunto de los asientos correspondientes a los participantes $\{0\}$, $\{0, 1, 2\}$, y $\{0, 1, 2, 3, 4, 5\}$ son rectangulares y son bellos. Así, la belleza de este mapa de asientos es 3, y `swap_seats` retorna 3.

Digamos que el grader llama `swap_seats(0, 5)` nuevamente. Luego de la petición 1, el mapa de asientos regresa al estado inicial. El conjunto de asientos correspondiente a los concursantes $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$, y $\{0, 1, 2, 3, 4, 5\}$ son rectangulares y bellos. Por lo tanto, la belleza del mapa de asientos es 4 y `swap_seats` debe retornar 4.

Los archivos `sample-01-in.txt` y `sample-01-out.txt` en el archivo zip adjunto corresponde a este ejemplo. Otros ejemplos de entrada/salida también se encuentran disponibles en el paquete.

Restricciones

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ para cualquier llamada a `swap_seats`
- $0 \leq b \leq HW - 1$ para cualquier llamada a `swap_seats`
- $a \neq b$ para cualquier llamada a `swap_seats`

Sub-tareas

1. (5 puntos) $HW \leq 100$, $Q \leq 5\,000$
2. (6 puntos) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 puntos) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 puntos) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ para cualquier llamada a `swap_seats`
5. (33 puntos) $H = 1$
6. (30 puntos) Sin restricciones adicionales

Grader de ejemplo

El grader de ejemplo lee la entrada en el formato siguiente:

- línea 1: $H W Q$
- línea $2 + i$ ($0 \leq i \leq HW - 1$): $R_i C_i$
- línea $2 + HW + j$ ($0 \leq j \leq Q - 1$): $A_j B_j$

Donde A_j y B_j son parámetros para la llamada a `swap_seats` para la petición j .

El grader de ejemplo imprime sus respuestas en el siguiente formato:

- línea $1 + j$ ($0 \leq j \leq Q - 1$): el valor de retorno de `swap_seats` para la petición j