



Seats

Você vai organizar uma competição internacional de programação numa sala de competição retangular, que tem HW assentos organizados em H linhas e W colunas. As linhas são numeradas de 0 a $H - 1$ e as colunas são numeradas de 0 a $W - 1$. O assento na linha r e coluna c é denotado por (r, c) . Você convidou HW competidores, numerados de 0 a $HW - 1$. Você também fez um mapa de assentos, que atribui o competidor i ($0 \leq i \leq HW - 1$) ao assento (R_i, C_i) . O mapa atribui exatamente um competidor a cada assento.

Um conjunto S de assentos na sala de competição é chamado de **retangular** se existem inteiros r_1, r_2, c_1 , e c_2 satisfazendo as seguintes condições:

- $0 \leq r_1 \leq r_2 \leq H - 1$.
- $0 \leq c_1 \leq c_2 \leq W - 1$.
- S é exatamente o conjunto de todos os assentos (r, c) tal que $r_1 \leq r \leq r_2$ e $c_1 \leq c \leq c_2$.

Um conjunto retangular de k assentos ($1 \leq k \leq HW$) é **belo** se os competidores cujos assentos estão no conjunto são numerados de 0 a $k - 1$. A **beleza** de um mapa de assentos é o número de conjuntos de assentos retangulares belos no mapa.

Depois de preparar o seu mapa de assentos, você recebe vários pedidos de troca de dois assentos atribuídos a dois competidores. Mais precisamente, existem Q pedidos numerados de 0 a $Q - 1$, em ordem cronológica. O pedido j ($0 \leq j \leq Q - 1$) é para trocar os assentos atribuídos aos competidores A_j e B_j . Você aceita cada pedido imediatamente e atualiza o mapa. Após cada atualização, seu objetivo é computar a beleza do mapa de assentos atual.

Detalhes de implementação

Você deve implementar o seguinte procedimento e função:

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

- H, W : o número de linhas e o número de colunas.
- R, C : arrays de comprimento HW representando o mapa de assentos inicial.
- Este procedimento é chamado exatamente uma vez, e antes de qualquer chamada a `swap_seats`.

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

- Esta função descreve um pedido para trocar dois assentos.
- a, b : competidores cujos assentos devem ser trocados.
- Esta função é chamada Q vezes.
- Esta função deve devolver a beleza do mapa de assentos depois da troca.

Exemplo

Seja $H = 2$, $W = 3$, $R = [0, 1, 1, 0, 0, 1]$, $C = [0, 0, 1, 1, 2, 2]$, e $Q = 2$.

O avaliador primeiramente chama `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])`.

No início, o mapa de assentos é o seguinte:

0	3	4
1	2	5

Digamos que o avaliador chama `swap_seats(0, 5)`. Depois do pedido 0, o mapa de assentos é o seguinte:

5	3	4
1	2	0

Os conjuntos de assentos correspondentes aos competidores $\{0\}$, $\{0, 1, 2\}$, e $\{0, 1, 2, 3, 4, 5\}$ são retangulares e belos. Assim, a beleza deste mapa de assentos é 3, e `swap_seats` deve retornar 3.

Digamos que o avaliador chama `swap_seats(0, 5)` novamente. Depois do pedido 1, o mapa de assentos retorna ao estado inicial. Os conjuntos de assentos correspondentes aos competidores $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$, e $\{0, 1, 2, 3, 4, 5\}$ são retangulares e belos. Portanto, a beleza deste mapa de assentos é 4, e `swap_seats` deve retornar 4.

Os ficheiros `sample-01-in.txt` e `sample-01-out.txt` do arquivo zip em anexo

correspondem a este exemplo. Outros inputs/outputs de exemplo também estão disponíveis no arquivo.

Restrições

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

Subtarefas

1. (5 pontos) $HW \leq 100$, $Q \leq 5\,000$
2. (6 pontos) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 pontos) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 pontos) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ para qualquer chamada a `swap_seats`
5. (33 pontos) $H = 1$
6. (30 pontos) Nenhuma restrição adicional

Avaliador de exemplo

O avaliador de exemplo lê a entrada no seguinte formato:

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

Aqui, A_j e B_j são parâmetros para a chamada a `swap_seats` para o pedido j .

O avaliador de exemplo imprime suas respostas no seguinte formato:

- linha $1 + j$ ($0 \leq j \leq Q - 1$): o valor de retorno de `swap_seats` para o pedido j .