



Места

Международно състезание по програмиране се провежда в правоъгълна зала, която разполага с $H \times W$ места, подредени в H реда и W колони. Редовете са номерирани с числата от 0 до $H - 1$, а колоните от 0 до $W - 1$. Мястото, намиращо се на пресечката на ред r и колона c , се отбелязва с (r, c) . Поканени за участие са $H \times W$ състезателя, номерирани от 0 до $H \times W - 1$. Изработена е схема на настаняване, в която за всеки състезател i ($0 \leq i \leq H \times W - 1$) е определено място (R_i, C_i) . Схемата съпоставя точно един състезател на всяко място.

Група места S се нарича **правоъгълна**, ако съществуват цели числа r_1, r_2, c_1 , и c_2 , които удовлетворяват следните условия:

- $0 \leq r_1 \leq r_2 \leq H - 1$.
- $0 \leq c_1 \leq c_2 \leq W - 1$.
- S е множеството от всички места (r, c) , такива че $r_1 \leq r \leq r_2$ и $c_1 \leq c \leq c_2$.

Правоъгълна група места, състояща се от k ($0 \leq k \leq H \times W$) места, се нарича **красива** ако състезателите, които са настанени в групата, имат номера от 0 до $k - 1$. **Красота** на схема на настаняване се нарича броят красиви правоъгълни групи места в цялата схема.

След приготвяне на схемата на настаняване пристигат заявки за размяна на местата на двойки състезатели. По-конкретно имаме Q такива заявки, номерирани от 0 до $Q - 1$ в хронологичен ред. Заявка с номер j ($0 \leq j \leq Q - 1$) е за размяна на състезатели с номера A_j и B_j . Всяка заявка води до незабавна промяна на схемата на настаняване. След всяка промяна, вашата задача е да изчислите **красотата** на текущата схема на настаняване.

Детайли за имплементацията

Трябва да напишете следните функции:

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

- H, W : броят редове и броят колони в залата.
- R, C : масиви с дължина $H \times W$ представляващи началната схема на настаняване.
- Тази процедура се извиква точно веднъж преди всички извиквания на

swap_seats.

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

- Тази функция описва заявка за размяна на местата на двама състезатели.
- a, b : номера на състезателите, които ще разменят местата си.
- Тази функция се извиква Q пъти.
- Функцията трябва да връща изчислената **красота** на схемата на настаняване след извършване на размяната.

Пример

Нека $H = 2, W = 3, R = [0, 1, 1, 0, 0, 1], C = [0, 0, 1, 1, 2, 2]$, и $Q = 2$.

Грейдърът първо ще извика `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])`.

Началната схема на настаняване изглежда така:

0	3	4
1	2	5

След това грейдърът извиква `swap_seats(0, 5)`. След заявка 0, схемата се променя до следната:

5	3	4
1	2	0

Множествата места, съответстващи на състезатели с номера $\{0\}$, $\{0, 1, 2\}$ и $\{0, 1, 2, 3, 4, 5\}$, са правоъгълни и красиви. Красотата на тази схема на настаняване е равна на 3 и съответно функцията `swap_seats` трябва да върне стойност 3.

Нека следващата заявка е отново `swap_seats(0, 5)`. След нея схемата на настаняване се връща в началния си вид. Множествата места, съответстващи на състезатели с номера $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$ и $\{0, 1, 2, 3, 4, 5\}$, са правоъгълни и

красиви. Следователно, красотата на схемата за настаняване е равна на 4 и функцията `swap_seats` трябва да върне стойност 4.

Файловете `sample-01-in.txt` и `sample-01-out.txt` в прикачения zip архив съответстват на горния пример. В архива има и други примерни тестови примери.

Ограничения

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ за всяко извикване на `swap_seats`
- $0 \leq b \leq HW - 1$ за всяко извикване на `swap_seats`
- $a \neq b$ за всяко извикване на `swap_seats`

Подзадачи

1. (5 точки) $HW \leq 100$, $Q \leq 5\,000$
2. (6 точки) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 точки) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 точки) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ за всяко извикване на `swap_seats`
5. (33 точки) $H = 1$
6. (30 точки) Без допълнителни ограничения

Примерен грейдър

Примерният грейдър чете входните данни в следния формат:

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

A_j и B_j са параметрите на извикването на `swap_seats` за заявка с номер j .

Примерният грейдър извежда вашите отговори в следния формат::

- ред $1 + j$ ($0 \leq j \leq Q - 1$) : стойността, върната от `swap_seats` за заявка с номер j