



Робочі місця

Ви збираєтесь влаштувати Міжнародні змагання з програмування в прямокутному приміщенні, в якому стоїть HW робочих місць (H рядів та W стовпців). Ряди пронумеровані від 0 до $H - 1$ а стовпці - від 0 до $W - 1$. Місце в ряді r та стовпці c позначається, як (r, c) . Ви запросили HW учасників, пронумерованих від 0 до $HW - 1$. Ви також зробили схему розсадки, що фіксує для учасника i ($0 \leq i \leq HW - 1$) місце (R_i, C_i) . Гарантовано, що за кожним місцем закріплено рівно одного учасника.

Множина місць S називається **прямокутною** якщо існують такі цілі числа r_1, r_2, c_1 , та c_2 , що задовольняють наступні умови:

- $0 \leq r_1 \leq r_2 \leq H - 1$.
- $0 \leq c_1 \leq c_2 \leq W - 1$.
- S множина всіх місць (r, c) таких, що $r_1 \leq r \leq r_2$ та $c_1 \leq c \leq c_2$.

Прямокутна множина, що складається з k місць ($1 \leq k \leq HW$) називається **красивою**, якщо учасники місця яких належать до цієї множини мають номери від 0 до $k - 1$. **Красою** схеми розсадки називається кількість красивих прямокутних множин в цій схемі.

Після підготовки своєї схеми розсадки, ви отримуєте кілька запитів від пар учасників поміняти місцями. Якщо бути точним, надходить Q таких запитів пронумерованих від 0 до $Q - 1$ в хронологічному порядку. Запит j ($0 \leq j \leq Q - 1$) покликаний поміняти місцями учасників A_j та B_j . Ви негайно приймаєте всі запити та вносите відповідні зміни в схему розсадки. Після кожної зміни ви маєте підрахувати красу поточної схеми розсадки.

Деталі реалізації

Ви маєте реалізувати наступні процедуру та функцію:

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

- H, W : кількість рядків і стовпців.
- R, C : масиви довжини HW , що відображають початкову розсадку.
- Ця процедура викликається рівно один раз і до всіх викликів `swap_seats`.

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

- Ця функція описує запит на зміну місцями.
- a, b : учасники, які хочуть поміняти місцями.
- Ця функція викликається Q разів.
- Ця функція повинна повернути красу схеми розсадки після зміни.

Приклад

Нехай $H = 2$, $W = 3$, $R = [0, 1, 1, 0, 0, 1]$, $C = [0, 0, 1, 1, 2, 2]$, та $Q = 2$.

Модуль перевірки спочатку викликає `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])`.

Спершу схема виглядає наступним чином.

0	3	4
1	2	5

Припустимо, що модуль перевірки викликає функцію `swap_seats(0, 5)`. Після запиту 0, схема розсадки буде виглядати так.

5	3	4
1	2	0

Множини місць, що відповідають учасникам $\{0\}$, $\{0, 1, 2\}$, та $\{0, 1, 2, 3, 4, 5\}$ прямокутні та красиві. Отже красою цієї схеми розсадки буде число 3, та `swap_seats` має повернути 3.

Припустимо, що модуль перевірки знову викликає `swap_seats(0, 5)`. Після запиту 1, схема розсадки повертається до початкового стану. Множини місць, що відповідають учасникам $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$, та $\{0, 1, 2, 3, 4, 5\}$ прямокутні та красиві. Тому красою цієї схеми буде число 4, та `swap_seats` має повернути 4.

Файли `sample-01-in.txt` та `sample-01-out.txt` ви знайдете у zip-архіві, що

додається. Інші приклади вхідних та вихідних даних також наявні у архіві.

Обмеження

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ для будь-якого виклику `swap_seats`
- $0 \leq b \leq HW - 1$ для будь-якого виклику `swap_seats`
- $a \neq b$ для будь-якого виклику `swap_seats`

Підзадачі

1. (5 балів) $HW \leq 100$, $Q \leq 5\,000$
2. (6 балів) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 балів) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 балів) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ для будь-якого виклику `swap_seats`
5. (33 бали) $H = 1$
6. (30 балів) без додаткових обмежень

Модуль перевірки з прикладу

Модуль перевірки з прикладу зчитує вхідні дані в наступному форматі:

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

Тут A_j та B_j - параметри для виклику функції `swap_seats` у запиті j .

Моуль перевірки із прикладу виводить вихідні дані у наступному форматі:

- рядок $1 + j$ ($0 \leq j \leq Q - 1$) : числа, що повернула функція `swap_seats` на запит j