



## 座席

あなたは国際プログラミングコンテストを、縦  $H$  行、横  $W$  列の  $HW$  個の座席のある長方形のホールで開く。行は  $0$  から  $H - 1$  までの番号がついており、列は  $0$  から  $W - 1$  までの番号がついている。 $r$  行  $c$  列にある座席は  $(r, c)$  と表記される。あなたは  $0$  から  $HW - 1$  までの番号のついた  $HW$  人の選手を招待した。また、選手  $i$  ( $0 \leq i \leq HW - 1$ ) を座席  $(R_i, C_i)$  に割り当てる座席表を作った。この表は各座席にちょうど  $1$  人の選手を割り当てる。

ホールの座席の集合  $S$  が長方形であるとは、以下を満たすような整数  $r_1, r_2, c_1, c_2$  が存在することである:

- $0 \leq r_1 \leq r_2 \leq H - 1$ .
- $0 \leq c_1 \leq c_2 \leq W - 1$ .
- $S$  は  $r_1 \leq r \leq r_2$  かつ  $c_1 \leq c \leq c_2$  を満たす座席  $(r, c)$  全体からなる集合である。

$k$  個 ( $1 \leq k \leq HW$ ) の座席からなる長方形の集合が美しいとは、それらの座席に割り当てられた生徒たちが  $0$  から  $k - 1$  までの番号をもつことである。座席表の美しさとは、その表での座席の集合のうち、長方形で美しいものの個数のことである。

あなたは座席表を準備したのち、 $2$  人の選手に割り当てられた座席を入れ替える要求をいくつか受け取った。詳しく言うと、時系列順に  $0$  から  $Q - 1$  までの番号がついた  $Q$  個の要求があり、要求  $j$  ( $0 \leq j \leq Q - 1$ ) は、選手  $A_j$  と  $B_j$  に割り当てられた座席の位置を入れ替えよというものである。あなたはそれぞれの要求をただちに受け入れ、表を更新する。それぞれの更新のあとで、その時点での座席表の美しさを計算するのが、あなたの仕事である。

## 実装の詳細

あなたは、以下のプロシージャと関数を実装する必要がある。

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

- $H, W$ : 行の数と列の数である。
- $R, C$ : 最初の座席表をあらわす長さ  $HW$  の配列である。
- このプロシージャが呼び出されるのはちょうど  $1$  回で、その時点で `swap_seats` はまだ  $1$  度も呼び出されていない。

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

- この関数は、 $2$  つの座席を入れ替える要求をあらわす。
- $a, b$ : 座席の位置を入れ替えられる選手の番号である。

- この関数は  $Q$  回呼び出される。
- この関数は、座席を入れ替えたあとの座席表の美しさを返す。

## 入出力例

$H = 2, W = 3, R = [0, 1, 1, 0, 0, 1], C = [0, 0, 1, 1, 2, 2], Q = 2$  とする。

採点プログラムはまず `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])` を呼び出す。

最初、座席表は以下の通りである。

0	3	4
1	2	5

ここで、採点プログラムが `swap_seats(0, 5)` を呼び出したとしよう。要求 0 のあと、座席表は以下のようになっている。

5	3	4
1	2	0

選手の集合  $\{0\}, \{0, 1, 2\}, \{0, 1, 2, 3, 4, 5\}$  に対応する座席の集合が長方形で美しい。ゆえに、この座席表の美しさは 3 であり、`swap_seats` は 3 を返すことになる。

さて、採点プログラムがまた `swap_seats(0, 5)` を呼び出したとしよう。要求 1 のあと、座席表は最初の状態に戻る。選手の集合  $\{0\}, \{0, 1\}, \{0, 1, 2, 3\}, \{0, 1, 2, 3, 4, 5\}$  に対応する座席の集合が長方形で美しい。ゆえに、座席表の美しさは 4 であり、`swap_seats` は 4 を返すことになる。

zip 圧縮された添付パッケージ (attachment package) に入ったファイル `sample-01-in.txt` および `sample-01-out.txt` がこの例に対応している。他の入出力例もこのパッケージで得られる。

## 制約

- $1 \leq H$ .
- $1 \leq W$ .
- $HW \leq 1\,000\,000$ .
- $0 \leq R_i \leq H - 1$  ( $0 \leq i \leq HW - 1$ ).
- $0 \leq C_i \leq W - 1$  ( $0 \leq i \leq HW - 1$ ).
- $(R_i, C_i) \neq (R_j, C_j)$  ( $0 \leq i < j \leq HW - 1$ ).
- $1 \leq Q \leq 50\,000$ .
- `swap_seats` の各呼び出しで  $0 \leq a \leq HW - 1$ .
- `swap_seats` の各呼び出しで  $0 \leq b \leq HW - 1$ .
- `swap_seats` の各呼び出しで  $a \neq b$ .

## 小課題

1. (5点)  $HW \leq 100$ ,  $Q \leq 5\,000$ .
2. (6点)  $HW \leq 10\,000$ ,  $Q \leq 5\,000$ .
3. (20点)  $H \leq 1\,000$ ,  $W \leq 1\,000$ ,  $Q \leq 5\,000$ .
4. (6点)  $Q \leq 5\,000$ , `swap_seats` の各呼び出しで  $|a - b| \leq 10\,000$ .
5. (33点)  $H = 1$ .
6. (30点) 追加の制約はない.

## 採点プログラムのサンプル

採点プログラムのサンプルの入力形式は以下の通りである.

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

ここで,  $A_j$  と  $B_j$  は要求  $j$  に対応する `swap_seats` の呼び出しの引数である.

採点プログラムのサンプルの出力形式は以下の通りである.

- $1 + j$  行目 ( $0 \leq j \leq Q - 1$ ): 要求  $j$  に対応する `swap_seats` の呼び出しの返り値.