



Sitzplätze

Du möchtest einen internationalen Programmierwettbewerb in einer rechteckigen Halle austragen. Die Halle hat HW Sitzplätze, welche in H Zeilen und W Spalten angeordnet sind. Die Zeilen sind von 0 bis $H - 1$, und die Spalten von 0 bis $W - 1$ durchnummeriert. Der Sitzplatz in Zeile r und Spalte c wird mit (r, c) bezeichnet. Du lädst HW Teilnehmer ein, welche von 0 bis $HW - 1$ durchnummeriert sind. Du hast bereits eine Sitzordnung vorbereitet, welche jedem Teilnehmer i ($0 \leq i \leq HW - 1$) einen Sitzplatz (R_i, C_i) zuweist. Die Sitzordnung weist jedem Teilnehmer genau einen Sitzplatz zu.

Wir nennen eine Menge S von Sitzplätzen in der Halle **rechteckig**, falls Ganzzahlen r_1 , r_2 , c_1 und c_2 existieren, sodass die folgenden Bedingungen erfüllt sind:

- $0 \leq r_1 \leq r_2 \leq H - 1$.
- $0 \leq c_1 \leq c_2 \leq W - 1$.
- S entspricht genau der Menge aller Sitzplätze (r, c) , sodass $r_1 \leq r \leq r_2$ und $c_1 \leq c \leq c_2$ gilt.

Eine rechteckige Menge bestehend aus k Sitzplätzen ($1 \leq k \leq HW$) nennen wir **wunderschön**, falls die Teilnehmer, welche zu Sitzplätzen in der Menge zugewiesen sind, genau die Zahlen von 0 bis $k - 1$ haben. Die **Schönheit** einer Sitzordnung ist die Anzahl wunderschöner rechteckiger Mengen von Sitzen in dieser Sitzordnung.

Nachdem du die Sitzordnung vorbereitet hast, erhältst du mehrere Anfragen, die zugewiesenen Sitzplätze von zwei Teilnehmern zu vertauschen. Genauer gesagt gibt es Q solcher Anfragen, welche in chronologischer Reihenfolge von 0 bis $Q - 1$ durchnummeriert sind. Die j -te Anfrage ($0 \leq j \leq Q - 1$) möchte die zugewiesenen Sitzplätze der Teilnehmer A_j und B_j vertauschen. Du akzeptierst jede Anfrage umgehend und aktualisierst die Sitzordnung. Nach jeder Aktualisierung sollst du die Schönheit der aktuellen Sitzordnung berechnen.

Implementationshinweise

Du sollst die folgenden Methoden und Funktionen implementieren:

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

- H, W : Die Anzahl Zeilen und Anzahl Spalten.

- R, C : Arrays der Länge HW , die ursprüngliche Sitzordnung.
- Diese Methode wird genau einmal aufgerufen, bevor `swap_seats` das erste Mal aufgerufen wird.

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

- Diese Funktion beschreibt eine Anfrage, zwei Sitzplätze zu vertauschen.
- a, b : Teilnehmer, deren Sitze vertauscht werden sollen.
- Diese Funktion wird Q Mal aufgerufen.
- Diese Funktion soll die Schönheit der Sitzordnung nach der Vertauschung zurück geben.

Beispiele

Sei $H = 2, W = 3, R = [0, 1, 1, 0, 0, 1], C = [0, 0, 1, 1, 2, 2]$ und $Q = 2$.

Der Grader ruft zuerst `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])` auf.

Zu Beginn schaut die Sitzordnung so aus.

0	3	4
1	2	5

Nehmen wir an, der Grader ruft `swap_seats(0, 5)` auf. Nach der Anfrage 0 schaut die Sitzordnung so aus.

5	3	4
1	2	0

Die Mengen der Sitzplätze, welche den Teilnehmern $\{0\}$, $\{0, 1, 2\}$ und $\{0, 1, 2, 3, 4, 5\}$ entsprechen, sind rechteckig und wunderschön. Demnach ist die Schönheit dieser Sitzordnung 3, und `swap_seats` sollte 3 zurück geben.

Nehmen wir an, der Grader ruft `swap_seats(0, 5)` erneut auf. Nach der Anfrage 1 ist

die Sitzordnung wieder identisch zum ursprünglichen Zustand. Die Mengen der Sitzplätze, welche den Teilnehmern $\{0\}$, $\{0,1\}$, $\{0,1,2,3\}$ und $\{0,1,2,3,4,5\}$ entsprechen, sind rechteckig und wunderschön. Daher ist die Schönheit dieser Sitzordnung 4, und `swap_seats` sollte 4 zurück geben.

Die Dateien `sample-01-in.txt` und `sample-01-out.txt` im Zip-Archiv entsprechen diesem Beispiel. Weitere Beispiele für Ein-/Ausgabe sind ebenfalls in diesem Archiv verfügbar.

Einschränkungen

- $1 \leq H, W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ für alle Aufrufe von `swap_seats`
- $0 \leq b \leq HW - 1$ für alle Aufrufe von `swap_seats`
- $a \neq b$ für alle Aufrufe von `swap_seats`

Teilaufgaben

1. (5 Punkte) $HW \leq 100$, $Q \leq 5\,000$
2. (6 Punkte) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 Punkte) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 Punkte) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ für alle Aufrufe von `swap_seats`
5. (33 Punkte) $H = 1$
6. (30 Punkte) Keine weiteren Einschränkungen

Beispielgrader

Der Beispielgrader liest von der Standardeingabe in folgendem Format:

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

Hier sind A_j und B_j Parameter für den Aufruf von `swap_seats` der j -ten Anfrage.

Der Beispielgrader gibt deine Antworten in folgendem Format aus:

- Zeile $1 + j$ ($0 \leq j \leq Q - 1$) : Der Rückgabewert von `swap_seats` für die j -te Anfrage.