



Seats

Πρόκειται να διοργανώσετε ένα διαγωνισμό προγραμματισμού σε μία ορθογώνια αίθουσα, η οποία έχει HW καθίσματα διατεταγμένα σε H σειρές και W στήλες. Οι σειρές είναι αριθμημένες από το 0 μέχρι το $H - 1$ και οι στήλες από το 0 μέχρι το $W - 1$. Το κάθισμα που βρίσκεται στη σειρά r και τη στήλη c συμβολίζεται ως (r, c) . Στον διαγωνισμό λαμβάνουν μέρος HW διαγωνιζόμενοι, αριθμημένοι από το 0 μέχρι το $HW - 1$. Έχετε φτιάξει ένα διάγραμμα θέσεων, στο οποίο ο διαγωνιζόμενος i ($0 \leq i \leq HW - 1$) βρίσκεται στο κάθισμα (R_i, C_i) . Το διάγραμμα τοποθετεί ακριβώς έναν διαγωνιζόμενο σε κάθε κάθισμα.

Ένα σύνολο καθισμάτων της αίθουσας S καλείται **ορθογώνιο** αν υπάρχουν ακεραίοι r_1, r_2, c_1 και c_2 , οι οποίοι να ικανοποιούν τις πιο κάτω συνθήκες:

- $0 \leq r_1 \leq r_2 \leq H - 1$.
- $0 \leq c_1 \leq c_2 \leq W - 1$.
- Το S είναι ακριβώς το σύνολο όλων των καθισμάτων (r, c) έτσι ώστε $r_1 \leq r \leq r_2$ και $c_1 \leq c \leq c_2$.

Ένα ορθογώνιο σύνολο αποτελούμενο από k καθίσματα ($1 \leq k \leq HW$), θεωρείται **όμορφο** αν στα καθίσματα που το αποτελούν κάθονται οι διαγωνιζόμενοι με αριθμούς από το 0 μέχρι το $k - 1$. Η **ομορφιά** ολόκληρου του διαγράμματος θέσεων καθορίζεται από το πλήθος των όμορφων ορθογώνιων συνόλων καθισμάτων που υπάρχουν μέσα στο διάγραμμα.

Μετά τη δημιουργία του διαγράμματος θέσεων, δέχεστε μερικά αιτήματα για να αντιμετωπίσετε ζεύγη διαγωνιζόμενων. Συγκεκριμένα, υπάρχουν Q τέτοια αιτήματα, αριθμημένα από το 0 μέχρι το $Q - 1$, σε χρονολογική σειρά. Το αίτημα j ($0 \leq j \leq Q - 1$) ζητά να αντιμετωπίσετε τους διαγωνιζόμενους με αριθμούς A_j and B_j . Αποδέχεστε αμέσως κάθε αίτημα αντιμετάθεσης και προχωράτε στην ενημέρωση του διαγράμματος. Μετά από κάθε ενημέρωση, στόχος σας είναι να υπολογίσετε την ομορφιά του υφιστάμενου διαγράμματος θέσεων.

Λεπτομέρειες υλοποίησης

Να υλοποιήσετε τις ακόλουθες συναρτήσεις:

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

- H, W : το πλήθος των σειρών και το πλήθος των στηλών.
- R, C : πίνακες μεγέθους HW που αναπαριστούν το αρχικό διάγραμμα θέσεων.
- Η συνάρτηση αυτή καλείται ακριβώς μία φορά, πριν από όλες τις κλήσεις της `swap_seats`.

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

- Αυτή η συνάρτηση περιγράφει ένα αίτημα αντιμετάθεσης δύο διαγωνιζομένων.
- a, b : οι δύο διαγωνιζόμενοι που πρόκειται να αντιμετατεθούν.
- Η συνάρτηση καλείται Q φορές.
- Η συνάρτηση πρέπει να επιστρέφει έναν ακέραιο: την ομορφιά του διαγράμματος θέσεων μετά την αντιμετάθεση.

Παράδειγμα

Ας υποθέσουμε ότι $H = 2, W = 3, R = [0, 1, 1, 0, 0, 1], C = [0, 0, 1, 1, 2, 2]$, και $Q = 2$.

Ο βαθμολογητής πρώτα καλεί την `give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])`.

Αρχικά, το διάγραμμα θέσεων είναι το πιο κάτω:

0	3	4
1	2	5

Έστω ότι ο βαθμολογητής καλεί την `swap_seats(0, 5)`. Μετά το αίτημα 0, το διάγραμμα θέσεων είναι το πιο κάτω:

5	3	4
1	2	0

Τα σύνολα των καθισμάτων που αντιστοιχούν στους διαγωνιζόμενους $\{0\}$, $\{0, 1, 2\}$, και $\{0, 1, 2, 3, 4, 5\}$ είναι ορθογώνια και όμορφα. Επομένως, η ομορφιά του διαγράμματος είναι 3 και η συνάρτηση `swap_seats` πρέπει να επιστρέψει 3.

Ας υποθέσουμε ότι ο βαθμολογητής καλεί την `swap_seats(0, 5)` ξανά. Μετά το αίτημα 1, το διάγραμμα θέσεων επιστρέφει στην αρχική του διάταξη. Τα σύνολα των καθισμάτων που αντιστοιχούν στους διαγωνιζόμενους $\{0\}$, $\{0, 1\}$, $\{0, 1, 2, 3\}$, και $\{0, 1, 2, 3, 4, 5\}$ είναι ορθογώνια και όμορφα. Επομένως, η ομορφιά του διαγράμματος είναι 4 και η συνάρτηση `swap_seats` πρέπει να επιστρέψει 4.

Τα αρχεία `sample-01-in.txt` και `sample-01-out.txt` που βρίσκονται στο συμπιεσμένο πακέτο, αντιστοιχούν σε αυτό το παράδειγμα. Στο πακέτο θα βρείτε και επιπρόσθετα παραδείγματα εισόδου και εξόδου.

Περιορισμοί

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$ ($0 \leq i \leq HW - 1$)
- $0 \leq C_i \leq W - 1$ ($0 \leq i \leq HW - 1$)
- $(R_i, C_i) \neq (R_j, C_j)$ ($0 \leq i < j \leq HW - 1$)
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$ για κάθε κλήση της `swap_seats`
- $0 \leq b \leq HW - 1$ για κάθε κλήση της `swap_seats`
- $a \neq b$ για κάθε κλήση της `swap_seats`

Υποπροβλήματα

1. (5 βαθμοί) $HW \leq 100$, $Q \leq 5\,000$
2. (6 βαθμοί) $HW \leq 10\,000$, $Q \leq 5\,000$
3. (20 βαθμοί) $H \leq 1\,000$, $W \leq 1\,000$, $Q \leq 5\,000$
4. (6 βαθμοί) $Q \leq 5\,000$, $|a - b| \leq 10\,000$ για κάθε κλήση της `swap_seats`
5. (33 βαθμοί) $H = 1$
6. (30 βαθμοί) Χωρίς επιπρόσθετους περιορισμούς

Υποδειγματικός βαθμολογητής

Ο υποδειγματικός βαθμολογητής διαβάζει την είσοδό του ως εξής:

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

Εδώ, τα A_j και B_j είναι οι παράμετροι της συνάρτησης `swap_seats` για το αίτημα j .

Ο υποδειγματικός βαθμολογητής τυπώνει τις απαντήσεις σας στην ακόλουθη μορφή:

- γραμμή $1 + j$ ($0 \leq j \leq Q - 1$): το αποτέλεσμα της `swap_seats` για το αίτημα j

