



## Seats

אתם עומדים לערוך תחרות תכנות בינלאומית באולם מלבני, שמכיל  $HW$  כיסאות (seats) המסודרים ב- $H$  שורות ו- $W$  עמודות. השורות ממוספרות מ-0 עד  $H - 1$  והעמודות ממוספרות מ-0 עד  $W - 1$ . הכיסא בשורה  $r$  ועמודה  $c$  מסומן ב- $(r, c)$ . הזמנתם  $HW$  מתחרים הממוספרים מ-0 עד  $HW - 1$ . הכנתם מפת ישיבה (seating chart), שמקצה את מתחרה  $i$  ( $0 \leq i \leq HW - 1$ ) לכיסא  $(R_i, C_i)$ . המפה מקצה בדיוק מתחרה אחד לכל כיסא.

קבוצת כיסאות  $S$  באולם נקראת **מלבנית** אם קיימים מספרים שלמים  $r_1, r_2, c_1, c_2$  המקיימים את התנאים הבאים:

- $0 \leq r_1 \leq r_2 \leq H - 1$
- $0 \leq c_1 \leq c_2 \leq W - 1$
- הקבוצה  $S$  היא בדיוק קבוצת כל הכיסאות  $(r, c)$  כך ש- $r_1 \leq r \leq r_2$  וגם  $c_1 \leq c \leq c_2$ .

קבוצה מלבנית של  $k$  כיסאות ( $1 \leq k \leq HW$ ) נקראת **יפה** אם מספרי המתחרים שהוקצו לכיסאות בקבוצה הם 0 עד  $k - 1$ . **היופי** של מפת ישיבה הוא מספר הקבוצות המלבניות היפות במפה.

לאחר שהכנתם את מפת הישיבה, אתם מקבלים מספר בקשות להחלפת שני מקומות שהוקצו לשני מתחרים. ליתר דיוק, יש  $Q$  בקשות הממוספרות מ-0 עד  $Q - 1$  בסדר כרונולוגי. הבקשה  $j$  ( $0 \leq j \leq Q - 1$ ) היא להחליף את המקומות של מתחרים  $A_j$  ו- $B_j$ . אתם מאשרים כל בקשה מיד ומעדכנים את המפה. לאחר כל עדכון, מטרתכם היא לחשב את היופי של מפת הישיבה הנוכחית.

## פרטי מימוש

עליכם לממש את הפונקציות הבאות:

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

- $H$  ו- $W$ : מספר השורות ומספר העמודות.
- $R$  ו- $C$ : מערכים באורך  $HW$  שמייצגים את מפת הישיבה ההתחלתית.
- פונקציה זו נקראת פעם אחת בלבד, לפני כל הקריאות ל-`swap_seats`.

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

- הפונקציה מתארת בקשה להחלפת מקומות בין שני מתחרים.
- $a, b$ : המתחרים שיש להחליף בין מקומותיהם.
- הפונקציה נקראת  $Q$  פעמים.
- הפונקציה צריכה להחזיר את היופי של מפת הישיבה לאחר ההחלפה.

## דוגמה

נתונים:

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

בהתחלה הגריידר קורא ל-

```
give_initial_chart(2, 3, [0, 1, 1, 0, 0, 1], [0, 0, 1, 1, 2, 2])
```

בהתחלה מפת הישיבה נראית כך:

0	3	4
1	2	5

נגיד שהגריידר קורא ל `swap_seats(0, 5)`. לאחר בקשה מספר 0 מפת הישיבה נראית כך:

5	3	4
1	2	0

קבוצות המושבים שמתאימות למתחרים  $\{0\}, \{0,1,2\}, \{0,1,2,3,4,5\}$  הן מלבניות ויפות. לכן היופי של מפת הישיבה הוא 3 ו-`swap_seats` צריכה להחזיר 3.

נגיד שהגריידר קורא ל `swap_seats(0, 5)` שוב. לאחר בקשה מספר 1, מפת הישיבה חוזרת למצב ההתחלתי שלה. קבוצות המושבים שמתאימות למתחרים  $\{0\}, \{0,1\}, \{0,1,2,3\}, \{0,1,2,3,4,5\}$  הן מלבניות ויפות. לכן היופי של מפת הישיבה הוא 4 ו-`swap_seats` תחזיר 4.

הקבצים `sample-01-in.txt` ו-`sample-01-out.txt` בקובץ ה-`zip` המצורף, מתאימים לדוגמה זו. קובץ ה-`zip` מכיל דוגמאות קלט/פלט נוספות.

## מגבלות

- $1 \leq H$
- $1 \leq W$
- $HW \leq 1\,000\,000$
- $0 \leq R_i \leq H - 1$  ( $0 \leq i \leq HW - 1$ )
- $0 \leq C_i \leq W - 1$  ( $0 \leq i \leq HW - 1$ )
- $(R_i, C_i) \neq (R_j, C_j)$  ( $0 \leq i < j \leq HW - 1$ )
- $1 \leq Q \leq 50\,000$
- $0 \leq a \leq HW - 1$  בכל קריאה ל-`swap_seats`.
- $0 \leq b \leq HW - 1$  בכל קריאה ל-`swap_seats`.
- $a \neq b$  בכל קריאה ל-`swap_seats`.

## נת משימות

1. (5 נקודות):  $HW \leq 100, Q \leq 5\,000$
2. (6 נקודות):  $HW \leq 10\,000, Q \leq 5\,000$
3. (20 נקודות):  $H \leq 1\,000, W \leq 1\,000, Q \leq 5\,000$
4. (6 נקודות):  $Q \leq 5\,000$  ו-  $|a - b| \leq 10\,000$  בכל קריאה ל-`swap_seats`.
5. (33 נקודות):  $H = 1$
6. (30 נקודות): ללא מגבלות נוספות.

## גריידר לדוגמה (Sample Grader)

הגריידר לדוגמה קורא קלט בפורמט הבא:

- שורה ראשונה:  $H$  ואחריו  $W$  ואחריו  $Q$ .
  - שורה מספר  $R_i : 2 + i$  ( $0 \leq i \leq HW - 1$ ) ואחריו  $C_i$
  - שורה מספר  $A_j : 2 + HW + j$  ( $0 \leq j \leq Q - 1$ ) ואחריו  $B_j$
- $A_j$  ו- $B_j$  הם הפרמטרים בקריאה ל-`swap_seats`, עבור בקשה מספר  $j$ .
- הגריידר לדוגמה מדפיס את תשובותיכם בפורמט הבא:
- שורה מספר  $1 + j$  ( $0 \leq j \leq Q - 1$ ): הערך שמחזירה `swap_seats` עבור בקשה מספר  $j$ .