



Rectangles

בתחילת המאה התשעה עשרה, השליט Hoseyngulu Khan Sardar ציווה לבנות ארמון ברמה הצופה לנהר יפה. נתאר את הרמה כטבלה של $n \times m$ תאים. שורות הטבלה ממוספרות מ-0 עד $n - 1$, והעמודות ממוספרות מ-0 עד $m - 1$.

נסמן ב- (i, j) את התא שבשורה i ובעמודה j ($0 \leq i \leq n - 1, 0 \leq j \leq m - 1$). לכל תא (i, j) יש גובה נתון המסומן $a[i][j]$.

השליט דרש מהאדריכלים שלו לבחור **אזור** מלבני לבניית הארמון. אסור שהאזור המלבני יכלול תאים שנמצאים על שפת הטבלה (שורה 0, שורה $n - 1$, עמודה 0, ועמודה $m - 1$). מכאן שעל האדריכלים לבחור ארבעה מספרים שלמים r_1, r_2, c_1, c_2 (כאשר $1 \leq r_1 \leq r_2 \leq n - 2$ ו- $1 \leq c_1 \leq c_2 \leq m - 2$), שמגדירים אזור הכולל בדיוק את התאים (i, j) המקיימים: $r_1 \leq i \leq r_2$ וגם $c_1 \leq j \leq c_2$.

בנוסף, אזור נחשב **מתאים**, אם ורק אם לכל תא (i, j) באזור מתקיים התנאי הבא:

- נתבונן בשני התאים הסמוכים לאזור ונמצאים בשורה i (התאים $(i, c_1 - 1)$ ו- $(i, c_2 + 1)$) וכן בשני התאים הסמוכים לאזור ונמצאים בעמודה j (התאים $(r_1 - 1, j)$ ו- $(r_2 + 1, j)$). אזי התא (i, j) חייב להיות נמוך ממש מכל ארבעת התאים לעיל.

משימתכם היא לעזור לאדריכלים למצוא את מספר האזורים המתאימים לארמון (כלומר מספר האפשרויות לבחור r_1, r_2, c_1, c_2 המגדירים אזור מתאים).

פרטי מימוש

עליכם לממש את הפונקציה הבאה:

```
int64 count_rectangles(int[][] a)
```

- a : מערך דו-מימדי של מספרים שלמים, בגודל n על m המייצג את גובהי התאים.
- על הפונקציה להחזיר את מספר האזורים המתאימים לארמון.

דוגמאות

דוגמה 1

הביטו בקריאה הבאה:

```
count_rectangles([[4, 8, 7, 5, 6],  
                 [7, 4, 10, 3, 5],  
                 [9, 7, 20, 14, 2],  
                 [9, 14, 7, 3, 6],  
                 [5, 7, 5, 2, 7],  
                 [4, 5, 13, 5, 6]])
```

4	8	7	5	6
7	4	10	3	5
9	7	20	14	2
9	14	7	3	6
5	7	5	2	7
4	5	13	5	6

יש 6 אזורים מתאימים, המתוארים להלן:

- $r_1 = r_2 = c_1 = c_2 = 1$
- $r_1 = 1, r_2 = 2, c_1 = c_2 = 1$
- $r_1 = r_2 = 1, c_1 = c_2 = 3$
- $r_1 = r_2 = 4, c_1 = 2, c_2 = 3$
- $r_1 = r_2 = 4, c_1 = c_2 = 3$
- $r_1 = 3, r_2 = 4, c_1 = c_2 = 3$

לדוגמה, $r_1 = 1, r_2 = 2, c_1 = c_2 = 1$, מגדירים אזור מתאים, כי מתקיימים התנאים הבאים:

- $a[1][1] = 4$ • קטן ממש מ- $a[0][1] = 8$, מ- $a[3][1] = 14$, מ- $a[1][0] = 7$ ומ- $a[1][2] = 10$.
- $a[2][1] = 7$ • קטן ממש מ- $a[0][1] = 8$, מ- $a[3][1] = 14$, מ- $a[2][0] = 9$ ומ- $a[2][2] = 20$.

מגבלות

- $1 \leq n, m \leq 2500$
- $0 \leq a[i][j] \leq 7000000$ (לכל $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$)

תת-משימות

1. $n, m \leq 30$ (8 נקודות)
2. $n, m \leq 80$ (7 נקודות)
3. $n, m \leq 200$ (12 נקודות)
4. $n, m \leq 700$ (22 נקודות)
5. $n \leq 3$ (10 נקודות)
6. $0 \leq a[i][j] \leq 1$ (לכל $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$) (13 נקודות)
7. (28 נקודות) ללא מגבלות נוספות.

גריידר לדוגמה

הגריידר לדוגמה קורא את הקלט בפורמט הבא (השורות נקראות משמאל לימין):

- שורה 1: $n \ m$
- שורה $i + 2$ (עבור $0 \leq i \leq n - 1$): $a[i][0] \ a[i][1] \ \dots \ a[i][m - 1]$

הגריידר מדפיס שורה אחת ובה הערך שהחזירה הפונקציה `count_rectangles`.