



Карнавални билети (tickets)

Ринго е на карнавал в Сингапур. В чантата си има няколко наградни билета, които би искал да използва на щанда с наградните игри. Всеки билет се предлага в един от n цвята и върху него е отпечатано цяло неотрицателно число. Целите числа, отпечатани на различни билети, може да са еднакви. Поради странността в карнавалните правила, за n е гарантирано, че е **четно**.

Ринго има в чантата си m билети за всеки цвят, т.е. има общо $n \cdot m$ билета. Върху билет j от цвят i има отпечатано число $x[i][j]$ ($0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$).

Играта с награди се играе в k кръга, номерирани от 0 до $k - 1$. Всеки кръг се играе в следния ред:

- От чантата си Ринго избира **комплект** от n билета, по един билет от всеки цвят. След това той дава комплекта на управителя на играта.
- Модераторът на играта си записва числата $a[0], a[1] \dots a[n - 1]$, отпечатани върху билетите от комплекта. Редът на тези n цели числа не е важен.
- Модераторът на играта изважда специална карта от щастливото поле за теглене и отбелязва цялото число b , отпечатано на специалната карта.
- Модераторът на играта пресмята абсолютните разлики между $a[i]$ и b за всяко i от 0 до $n - 1$. Нека S да е сумата от тези абсолютни разлики.
- За този кръг модераторът на играта дава на Ринго награда със стойност, равна на S .
- Билетите в комплекта се изхвърлят и не могат да се използват в бъдещи кръгове.

Останалите билети в чантата на Ринго след k кръга в играта се изхвърлят.

Като наблюдаваше внимателно, Ринго осъзна, че играта с награди е подправена! Вътре в щастливата кутия има принтер. Във всеки кръг модераторът на играта намира цяло число b , което минимизира стойността на наградата от този кръг. Стойността, избрана от модератора на играта, се отпечатва на специалната карта за този кръг.

Разполагайки с цялата тази информация, Ринго би искал да разпредели билети за кръговете на играта. Т.е. той иска да избере комплект от билети, който да се използва във всеки кръг, с цел да максимизира общата стойност на наградите.

Имплементация

Трябва да напишете следната процедура:

```
int64 find_maximum(int k, int[][] x)
```

- k : брой кръгове в играта.
- x : масив от $n \times m$ числа, описващ числата на билетите. Билетите от всеки цвят са сортирани в ненамаляващ ред на техните цели числа.
- Тази процедура ще бъде извикана точно веднъж.
- Тази процедура трябва да направи точно едно извикване на `allocate_tickets` (виж по-долу), чрез k комплекта от билети, по един за всеки кръг. Разпределението в комплектите трябва максимизира общата стойност на наградите.
- Тази процедура трябва да върне максималната обща стойност на наградите.

Процедурата `allocate_tickets` е дефинирана по следния начин:

```
void allocate_tickets(int[][] s)
```

- s : масив от $n \times m$. Стойността на $s[i][j]$ трябва да бъде r , ако билетът j от цвета i се използва в комплекта от кръг r на играта, или -1 , ако не се използва.
- За всяко $0 \leq i \leq n - 1$, измежду $s[i][0], s[i][1], \dots, s[i][m - 1]$, всяка стойност $0, 1, 2, \dots, k - 1$ трябва да се появи точно веднъж, а останалите стойности трябва да бъдат -1 .
- Ако има няколко разпределения, водещи до максимална обща стойност на наградата, вие може да използвате кое да е от тях.

Примери

Пример 1

Разглеждаме следното извикване:

```
find_maximum(2, [[0, 2, 5], [1, 1, 3]])
```

Това означава, че:

- има $k = 2$ кръга;
- целите числа, отпечатани на билетите с цвят 0, са съответно 0, 2 и 5;
- целите числа, отпечатани на билетите с цвят 1, са съответно 1, 1 и 3.

Възможно разпределение, което дава максималната обща стойност на наградата, е:

- В кръг 0, Ринго избира билет 0 от цвят 0 (с цялото число 0) и билет 2 от цвят 1 (с цялото число 3). Най-ниската възможна стойност на наградата в този кръг е 3. Например, управителят на играта може да избере $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- В кръг 1, Ринго избира билет 2 от цвят 0 (с цялото число 5) и билет 1 за цвят 1 (с цялото число 1). Най-ниската възможна стойност на наградата в този кръг е 4. Например, управителят на играта може да избере $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Следователно общата стойност на наградите ще бъде $3 + 4 = 7$.

За да посочи това разпределение, процедурата `find_maximum` трябва да направи следното извикване на `allocate_tickets`:

- `allocate_tickets ([[0, -1, 1], [-1, 1, 0]])`

И накрая, процедурата `find_maximum` трябва да върне 7.

Пример 2

Разглеждаме следното извикване:

```
find_maximum(1, [[5, 9], [1, 4], [3, 6], [2, 7]])
```

Това означава, че:

- има само един кръг,
- целите числа, отпечатани на билетите с цвят 0, са съответно 5 и 9;
- целите числа, отпечатани на билетите с цвят 1, са съответно 1 и 4;
- целите числа, отпечатани на билетите с цвят 2, са съответно 3 и 6;
- целите числа, отпечатани на билетите с цвят 3, са съответно 2 и 7.

Възможно разпределение, което дава максималната обща стойност на наградата, е:

- В кръг 0, Ринго избира билет 1 от цвят 0 (с цялото число 9), билет 0 от цвят 1 (с цялото число 1), билет 0 от цвят 2 (с цялото число 3) и билет 1 от цвят 3 (с цялото число 7). Най-ниската възможна стойност на наградата в този кръг е 12, когато управителят на играта избере $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

За да посочи това решение, процедурата `find_maximum` трябва да направи следното извикване на `allocate_tickets`:

- `allocate_tickets ([[-1, 0], [0, -1], [0, -1], [-1, 0]])`

И накрая, процедурата `find_maximum` трябва да върне 12.

Ограничения

- $2 \leq n \leq 1500$ и n е четно.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (за всяко $0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (за всяко $0 \leq i \leq n - 1$ и $1 \leq j \leq m - 1$)

Подзадачи

1. (11 точки) $m = 1$
2. (16 точки) $k = 1$

3. (14 точки) $0 \leq x[i][j] \leq 1$ (за всяко $0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$)
4. (14 точки) $k = m$
5. (12 точки) $n, m \leq 80$
6. (23 точки) $n, m \leq 300$
7. (10 точки) Без допълнителни ограничения.

Примерен грейдер

Примерният грейдер чете входните данни в следния формат:

- ред 1: $n \ m \ k$
- ред $2 + i$ ($0 \leq i \leq n - 1$): $x[i][0] \ x[i][1] \ \dots \ x[i][m - 1]$

Примерният грейдер отпечатва отговора в следния формат:

- ред 1: върнатата стойност от `find_maximum`
- ред $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$