



Karnaval chiptalari (tickets)

Ringo Singapurdagi karnavalda. Uni sumkasida sovrinli chiptalar bo'lib, ularni sovrin nomli o'yinda ishlatmoqchi. Har bir chipta n xil rangdan biriga ega bo'lib, unda nomanfiy butun son yozilgan. Turli xil chiptadagi sonlar bir xil bo'lishi mumkin. Karnaval qoidalariga ko'ra, n har doim **juft** bo'ladi.

Ringoda har bir rang uchun m tadan chipta bor, ya'ni $n \cdot m$ ta chipta. i rangli j -chiptaga $x[i][j]$ soni yozilgan ($0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$).

Prize o'yini k ta roundda o'ynaladi va ular 0 dan $k - 1$ gacha raqamlangan. Har bir round quyidagi tartibda o'ynaladi:

- Ringo sumkasidan har bitta rangdan bittadan olib n ta chiptalar **to'plamini** oladi. So'ngra uni o'yin hakamiga beradi.
- Hakam to'plamdagi chiptalarda yozilgan sonlarni $a[0], a[1] \dots a[n - 1]$ tarzda yozib oladi. Bu sonlarning tartibi muhim emas.
- Hakam omadli qutidan b butun soni yozilgan maxsus kartani oladi.
- Hakam 0 dan $n - 1$ gacha har bir i uchun $a[i]$ va b sonlarini absolyut farqini oladi. Aytaylik, S shu sonlar yig'indisi bo'lsin.
- Shu round uchun hakam Ringoga S ochko beradi.
- To'plamda ishlatilgan chiptalar olib tashlanadi va qayta ishlatib bo'lmaydi.

k ta rounddan keyingi Ringoda qolgan chiptalar ham olib tashlanadi.

Ringo diqqat bilan kuzatib, sovrin o'yini g'irromligini sezib qoldi! Aslida omadli qutini ichida printer bor ekan. Har bir roundda, hakam sovrinni qiymati eng kichik bo'ladigan b sonini tanlar ekan. So'ngra hakam tanlagan son maxsus kartaga chiqarilar ekan.

Shu ma'lumotlar orqali, Ringo chiptalarni roundlar bo'yicha taqsimlamoqchi.

Ya'ni har bir roundda umumiy sovrinni maksimallashtirishradigan chiptalar to'plamini tanlamoqchi.

Tafsilotlar

Quyidagi funktsiyani bajarishingiz lozim:

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

- k : roundlar soni.
- x : har bir chiptani ifodalovchi $n \times m$ masiv. Har bir rangdagi chiptalar qiymatlari bo'yicha kamaymaydigan tartibda saralangan.

- Bu funksiya aniq bir marta chaqiriladi.
- Bu funksiya har bir rounddagi chiptalarni ifodalash uchun `allocate_tickets`(pastda keltirilgan) funksiyasini aniq bir marta chaqirishi lozim. Chiptalarni taqsimlashda sovrinni maksimallashtirish lozim.
- Bu funksiya maksimum umumiy sovrinni qaytarishi lozim.

`allocate_tickets` funksiyasi quyidagicha:

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

- $s: n \times m$ massiv. Agar i -rangli j -chipta r -roundda ishlatilgan bo'lsa $s[i][j]$ ni qiyamti r bo'lishi lozim, agar ishlatilmasa -1 .
- Har bir $0 \leq i \leq n - 1$ uchun, $s[i][0], s[i][1], \dots, s[i][m - 1]$ da $0, 1, 2, \dots, k - 1$ sonlari bir marta uchrasi lozim va qolgan barchasi -1 bo'lishi lozim.
- If there are multiple allocations resulting in the maximum total prize value, it is allowed to report any of them.

Examples

Example 1

Consider the following call:

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

This means that:

- there are $k = 2$ rounds;
- the integers printed on the tickets of colour 0 are 0, 2 and 5, respectively;
- the integers printed on the tickets of colour 1 are 1, 1 and 3, respectively.

A possible allocation that gives the maximum total prize value is:

- In round 0, Ringo picks ticket 0 of colour 0 (with the integer 0) and ticket 2 of colour 1 (with the integer 3). The lowest possible value of the prize in this round is 3. E.g., the game master may choose $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- In round 1, Ringo picks ticket 2 of colour 0 (with the integer 5) and ticket 1 of colour 1 (with the integer 1). The lowest possible value of the prize in this round is 4. E.g., the game master may choose $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Therefore, the total value of the prizes would be $3 + 4 = 7$.

To report this allocation, the procedure `find_maximum` should make the following call to `allocate_tickets`:

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

Finally, the procedure `find_maximum` should return 7.

Example 2

Consider the following call:

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

This means that:

- there is only one round,
- the integers printed on the tickets of colour 0 are 5 and 9, respectively;
- the integers printed on the tickets of colour 1 are 1 and 4, respectively;
- the integers printed on the tickets of colour 2 are 3 and 6, respectively;
- the integers printed on the tickets of colour 3 are 2 and 7, respectively.

A possible allocation that gives the maximum total prize value is:

- In round 0, Ringo picks ticket 1 of colour 0 (with the integer 9), ticket 0 of colour 1 (with the integer 1), ticket 0 of colour 2 (with the integer 3), and ticket 1 of colour 3 (with the integer 7). The lowest possible value of the prize in this round is 12, when the game master chooses $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

To report this solution, the procedure `find_maximum` should make the following call to `allocate_tickets`:

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

Finally, the procedure `find_maximum` should return 12.

Constraints

- $2 \leq n \leq 1500$ and n is even.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (for all $0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (for all $0 \leq i \leq n - 1$ and $1 \leq j \leq m - 1$)

Subtasks

1. (11 points) $m = 1$
2. (16 points) $k = 1$
3. (14 points) $0 \leq x[i][j] \leq 1$ (for all $0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$)
4. (14 points) $k = m$
5. (12 points) $n, m \leq 80$
6. (23 points) $n, m \leq 300$

7. (10 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

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

The sample grader prints your answer in the following format:

- line 1: the return value of `find_maximum`
- line $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$