Tickets de carnaval (tickets)

Ringo está en un carnaval en Singapur. él tiene algunos tickets de premios en su mochila que quiere usar en el puesto de juego de premios. Cada ticket viene en uno de n colores y tiene un entero no negativo impreso en él. Los enteros impresos en distintos tickets pueden ser iguales. Debido a una rareza de las reglas del carnaval, se garantiza que n es **par**.

Ringo tiene m tickets de cada color en su mochila, lo que da un total de $n \cdot m$ tickets. El ticket j del color i tiene el entero $x[i][j]$ impreso en el ($0 \leq i \leq n-1$ y $0 \leq j \leq m-1$).

El juego de premios se juega en k rondas, numeradas de 0 a $k-1$. Cada ronda se juega en el siguiente orden:

- De su mochila, Ringo selecciona un **conjunto** de n tickets, un ticket de cada color. Luego le da el conjunto al árbitro del juego.
- El árbitro del juego anota los enteros $a[0]$, $a[1]$ $\ldots$ $a[n-1]$ impresos en los tickets del conjunto. El orden de estos n enteros no es importante.
- El árbitro del juego saca una carta especial de la caja de sorteo de la suerte y anota el entero b impreso en la carta.
- El árbitro del juego calcula las diferencias absolutas entre $a[i]$ y b para cada i de 0 a $n-1$. Sea S la suma de estas diferencias absolutas.
- Para esta ronda, el árbitro del juego le da a Ringo un premio con un valor de S.
- Se descartan los tickets en el conjunto y los mismos no pueden utilizarse en futuras rondas.

Se descartan los tickets restantes en la mochila de Ringo después de k rondas del juego.

Al observar de cerca, Ringo se da cuenta de que °el juego de premios es fraudulento! En realidad hay una impresora dentro de la caja de sorteo de la suerte. En cada ronda, el árbitro del juego encuentra un entero b que minimiza el valor del premio en esa ronda. El valor escogido por el árbitro del juego se imprime en la carta especial para esa ronda.

Teniendo toda esta información, Ringo quiere asignar tickets a las rondas del juego. Es decir, él quiere seleccionar el conjunto de tickets a usar en cada ronda que maximice el valor total de los premios.

Detalles de implementación

Deberás implementar la siguiente función:

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

- k: El número de rondas.
- x: un arreglo de tamaño $n \times m$ que describe los enteros en cada ticket. Los tickets de cada color están ordenados en orden no decreciente de sus enteros.
- Esta función se llama exactamente una vez.
- Esta función deberìa hacer exactamente un llamado a `allocate_tickets` (ver abajo) describiendo k conjuntos de tickets, uno para cada ronda. La asignación debe maximizar el valor total de los premios.
- Esta función debe retornar el máximo valor total de los premios.

La función `allocate_tickets` está definida de la siguiente forma:

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

- s: Un arreglo de $n \times m$. El valor de $s[i][j]$ debe ser r si el ticket j del color i se usa en el conjunto de la ronda r del juego, o -1 si no se usa nunca.
- Para cada $0 \leq i \leq n-1$, entre los n˙meros $s[i][0], s[i][1], \ldots, s[i][m-1]$ cada valor $0, 1, 2, \ldots, k-1$ debe ocurrir exactamente una vez, y todas las restantes posiciones del arreglo deben ser -1.
- Si hay múltiples asignaciones que resultan en el máximo valor total de premios, se permite reportar cualquiera de ellas.

Ejemplos

Ejemplo 1

Considere el siguiente llamado:

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

Esto significa que:

- Hay $k = 2$ rondas;
- Los enteros impresos en los tickets de color 0 son 0, 2 y 5, respectivamente;
- Los enteros impresos en los tickets de color 1 son 1, 1 y 3, respectivamente.

Una posible asignación que da el máximo valor total de premios es:

- En la ronda 0, Ringo elige el ticket 0 del color 0 (con el entero 0) y el ticket 2 del color 1 (con el entero 3). El menor valor posible para el premio en esta ronda es 3. Por ejemplo, el árbitro del juego puede elegir $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- En la ronda 1, Ringo elige el ticket 2 del color 0 (con el entero 5) y el ticket 1 del color 1 (con el entero 1). El menor valor posible para el premio en esta ronda es 4. Por ejemplo, el árbitro del juego puede elegir $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Entonces el valor total de los premios serìa de $3 + 4 = 7$.

Para reportar esta asignación, la función `find_maximum` debe hacer el siguiente llamado a `allocate_tickets`:

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

Finalmente, la función `find_maximum` debe retornar 7.

Ejemplo 2

Considere el siguiente llamado:

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

Esto significa que:

- Solo hay una ronda,
- Los enteros impresos en los tickets de color 0 son 5 y 9, respectivamente;
- Los enteros impresos en los tickets de color 1 son 1 y 4, respectivamente;
- Los enteros impresos en los tickets de color 2 son 3 y 6, respectivamente;
- Los enteros impresos en los tickets de color 3 son 2 y 7, respectivamente.

Una posible asignación que da el máximo valor total de premios es:

- En la ronda 0, Ringo elige el ticket 1 del color 0 (con el entero 9), el ticket 0 del color 1 (con el entero 1), el ticket 0 del color 2 (con el entero 3), y el ticket 1 del color 3 (con el entero 7). El menor valor posible del premio en esta ronda es 12, cuando el árbitro del juego elige $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Para reportar esta solución, la función `find_maximum` debe hacer el siguiente llamado a `allocate_tickets`:

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

Finalmente, la función `find_maximum` debe retornar 12.

Cotas

- $2 \leq n \leq 1500$ y n es par.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (para todo $0 \leq i \leq n - 1$ y $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (para todo $0 \leq i \leq n - 1$ y $1 \leq j \leq m - 1$)

Subtareas

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

3. (14 puntos) $0 \leq x[i][j] \leq 1$ (para todo $0 \leq i \leq n-1$ y $0 \leq j \leq m-1$)
4. (14 puntos) $k = m$
5. (12 puntos) $n, m \leq 80$
6. (23 puntos) $n, m \leq 300$
7. (10 puntos) Sin restricciones adicionales.

Evaluador Local

El evaluador local lee la entrada con el siguiente formato:

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

El evaluador local imprime tu respuesta con el siguiente formato:

- línea 1: lo que devuelve `find_maximum`
- línea $2 + i$ ($0 \leq i \leq n-1$): $s[i][0]$ $s[i][1]$ $\ldots$ $s[i][m-1]$