

Tickets de Carnaval (tickets)

Ringo está en un carnaval en Singapur. Él tiene algunos tickets para premios en su morral, y quiere usarlos en el puesto de premios. Cada ticket viene en uno de n colores y tiene un entero no negativo impreso en él. Los enteros impresos en tickets distintos pueden ser iguales. Por una peculiaridad de las reglas del carnaval, está garantizado que n es **par**.

Ringo tiene m tickets de cada color en su morral, es decir, un total de $n \cdot m$ tickets. El ticket j de color i tiene el entero $x[i][j]$ impreso en él ($0 \leq i \leq n - 1$ y $0 \leq j \leq m - 1$).

El juego para obtener premios es jugado en k rondas, numeradas desde 0 hasta $k - 1$. Cada ronda es jugada en el siguiente orden:

- De su morral, Ringo selecciona un **conjunto** de n tickets, un ticket de cada color. Luego le da este conjunto al maestro del juego.
- El maestro anota los enteros $a[0], a[1] \dots a[n - 1]$ impresos en los tickets del conjunto. El orden de estos n enteros no es importante.
- El maestro saca una tarjeta de una caja de la suerte y anota el entero b escrito en la tarjeta.
- El maestro calcula la diferencia absoluta entre $a[i]$ y b para cada i desde 0 hasta $n - 1$. Sea S la suma de estas diferencias absolutas.
- Por esta ronda, el maestro le da a Ringo un premio de valor igual a S .
- Los tickets en el conjunto son descartados y no pueden ser usados en rondas futuras.

Los tickets restantes en el morral de Ringo después de las k rondas del juego son descartados.

Mirando detalladamente, ¡Ringo se dio cuenta que el juego está arreglado! De hecho hay una impresora dentro de la caja de la suerte. En cada ronda, el maestro encuentra el entero b que minimiza el valor del premio de esa ronda. El valor escogido por el maestro es impreso en la tarjeta especial de esa ronda.

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

Detalles de implementación

Debes implementar el siguiente procedimiento:

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

- k : el número de rondas.

- x : un arreglo $n \times m$ que describe los enteros en cada ticket. Los tickets del mismo color están ordenados de forma no decreciente según los enteros impresos en ellos.
- Este procedimiento es llamado exactamente una vez.
- Este procedimiento debe hacer exactamente un llamado a `allocate_tickets` (ver debajo), describiendo k conjuntos de tickets, uno para cada ronda. La asignación debe maximizar el valor total de los premios.
- Este procedimiento debe retornar el valor total máximo de los premios.

El procedimiento `allocate_tickets` es definido como sigue:

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

- s : un arreglo $n \times m$. El valor de $s[i][j]$ debe ser r si el ticket j de color i es usado en el conjunto de la ronda r del juego, o -1 si no es usado en ninguna ronda.
- Para cada $0 \leq i \leq n - 1$, en $s[i][0], s[i][1], \dots, s[i][m - 1]$ cada valor $0, 1, 2, \dots, k - 1$ debe ocurrir exactamente una vez, y todos los demás valores deben ser -1 .
- Si hay multiples asignaciones que resultan en el valor total máximo de los premios, está permitido reportar cualquiera de ellas.

Ejemplos

Ejemplo 1

Considera la siguiente llamada:

```
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 and 3, respectivamente.

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

- En la ronda 0, Ringo selecciona el ticket 0 de color 0 (con el entero 0) y el ticket 2 de color 1 (con el entero 3). El valor más pequeño posible para el premio de esta ronda es 3. Por ejemplo, el maestro podría escoger $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- En la ronda 1, Ringo selecciona el ticket 2 de color 0 (con el entero 5) y el ticket 1 de color 1 (con el entero 1). El valor más pequeño posible para el premio de esta ronda es 4. Por ejemplo, el maestro podría escoger $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Por lo tanto, el valor total de los premios sería $3 + 4 = 7$.

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

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

Finalmente, el procedimiento `find_maximum` debe retornar 7.

Ejemplo 2

Considera la siguiente llamada:

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

Esto significa que:

- hay una sola 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 obtiene el valor total de premios máximo es:

- En la ronda 0, Ringo selecciona el ticket 1 de color 0 (con el entero 9), el ticket 0 de color 1 (con el entero 1), el ticket 0 de color 2 (con el entero 3), y ticket 1 de color 3 (con el entero 7). El valor más pequeño posible para el premio de esta ronda es 12, cuando el maestro escoge $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

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

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

Finalmente, el procedimiento `find_maximum` debe retornar 12.

Restricciones

- $2 \leq n \leq 1500$ 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 de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

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

El evaluador de ejemplo imprime tu respuesta en el siguiente formato:

- línea 1: el valor de retorno de `find_maximum`
- línea $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$