

Losy (tickets)

Řehoř si na pouti v Singapuru nakoupil losy do loterie. Každý z losů má jednu z n barev, kde n je **sudé**, a je na něm napsáno nezáporné celé číslo. Na různých losech může být napsané stejné číslo.

Řehoř si od každé z barev koupil m losů, má jich tedy dohromady $n \cdot m$. Na j -tém losu barvy i je celé číslo $x[i][j]$ ($0 \leq i \leq n - 1$ a $0 \leq j \leq m - 1$).

Loterie má k kol, očíslovaných od 0 do $k - 1$. Každé kolo probíhá následovně:

- Řehoř si ze svých losů vybere **množinu** velikosti n , obsahující právě jeden los od každé barvy. Tuto množinu poté předá losujícímu.
- Losující si poznamená čísla $a[0], a[1] \dots a[n - 1]$ napsaná na losech z této množiny; pořadí těchto čísel není důležité.
- Losující vytáhne lístek z mysteriózní krabice a zapíše si celé číslo b na něm vytištěné.
- Pro i od 0 do $n - 1$ losující spočítá absolutní hodnotu rozdílu čísel $a[i]$ a b . Jako S označme součet těchto absolutních hodnot.
- Losující vyplatí Řehořovi S Singapurských dolarů.
- Losující poté zničí losy z tohoto kola, Řehoř je tedy nemůže použít v následujících kolech.

Po k kolech hra končí a zbylé losy Řehoř vyhodí.

Řehoř si ale všiml, že losující podvádí! V mysteriózní krabici je tiskárna, která v každém kole vytiskne takové celé číslo b , pro které Řehoř v tomto kole (na základě hodnot $a[0], a[1] \dots a[n - 1]$) získá nejméně dolarů.

Řehoř chce vymyslet, jak si v jednotlivých kolech loterie volit losy tak, aby celkově získal co nejvíc dolarů.

Implementační detaily

Implementujte následující funkci:

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

- k : počet kol.
- x : pole o rozměrech $n \times m$ udávající čísla na jednotlivých losech. V rámci každé barvy jsou čísla na losech seříděna v neklesajícím pořadí.
- Tato funkce bude zavolána právě jednou.
- Tato funkce musí právě jednou zavolat funkci `allocate_tickets` popsanou níže, s jejímž

použitím oznámí k množin losů, jednu pro každé kolo loterie. Tyto množiny musí maximalizovat počet dolarů, které Řehoř celkem získá za celou loterii.

- Tato funkce musí vrátit počet dolarů, které Řehoř získá.

Funkce `allocate_tickets` má následující deklaraci:

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

- s : pole o rozměrech $n \times m$. Hodnota $s[i][j]$ musí být r , jestliže Řehoř použije j -tý los barvy i v kole r , nebo -1 v případě, že tento los nepoužije.
- Pro každé i tž. $0 \leq i \leq n - 1$ se musí každé z čísel $0, 1, 2, \dots, k - 1$ vyskytovat v posloupnosti $s[i][0], s[i][1], \dots, s[i][m - 1]$ právě jednou a všechna ostatní čísla v této posloupnosti musí být -1 .
- Jestliže existuje více možných výběrů losů, které maximalizují počet Řehořem získaných dolarů, zavolejte tuto funkci pro jeden libovolný z nich.

Příklady

Příklad 1

Uvažme následující volání:

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

Tedy:

- jsou $k = 2$ kola;
- na losech barvy 0 jsou čísla 0, 2 a 5;
- na losech barvy 1 jsou čísla 1, 1 a 3.

Jedno z možných řešení je následující:

- V kole 0 Řehoř vybere 0-tý los barvy 0 (na němž je napsáno číslo 0) a 2-hý los barvy 1 (na němž je napsáno číslo 3). Nejmenší možný počet vyplacených dolarů je 3, čehož losující dosáhne například volbou $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- V kole 1 Řehoř vybere 2-hý los barvy 0 (na němž je napsáno číslo 5) a 1-ní los barvy 1 (na němž je napsáno číslo 1). Nejmenší možný počet vyplacených dolarů je 4, čehož losující dosáhne například volbou $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Řehoř tedy celkem získá $3 + 4 = 7$ dolarů.

Aby nahlásila tuto volbu losů, funkce `find_maximum` zavolá funkci `allocate_tickets` s následujícími parametry:

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

Funkce `find_maximum` poté vrátí hodnotu 7.

Příklad 2

Uvažme následující volání:

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

Tedy:

- je jen jedno kolo,
- na losech barvy 0 jsou čísla 5 a 9;
- na losech barvy 1 jsou čísla 1 a 4;
- na losech barvy 2 jsou čísla 3 a 6;
- na losech barvy 3 jsou čísla 2 a 7.

Jedno z možných řešení je následující:

- V kole 0 Řehoř vybere 1-ní los barvy 0 (na němž je napsáno číslo 9), 0-tý los barvy 1 (na němž je napsáno číslo 1), 0-tý los barvy 2 (na němž je napsáno číslo 3) a 1-ní los barvy 3 (na němž je napsáno číslo 7).
- Nejmenší možný počet vyplacených dolarů je 12, čehož losující dosáhne například volbou $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Aby nahlásila tuto volbu losů, funkce `find_maximum` zavolá funkci `allocate_tickets` s následujícími parametry:

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

Funkce `find_maximum` poté vrátí hodnotu 12.

Omezení

- $2 \leq n \leq 1500$ a n je sudé.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (pro všechna i, j tž. $0 \leq i \leq n - 1$ a $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (pro všechna i, j tž. $0 \leq i \leq n - 1$ a $1 \leq j \leq m - 1$)

Podúlohy

1. (11 bodů) $m = 1$
2. (16 bodů) $k = 1$
3. (14 bodů) $0 \leq x[i][j] \leq 1$ (pro všechna i, j tž. $0 \leq i \leq n - 1$ a $0 \leq j \leq m - 1$)
4. (14 bodů) $k = m$
5. (12 bodů) $n, m \leq 80$
6. (23 bodů) $n, m \leq 300$
7. (10 bodů) Bez dalších omezení.

Ukázkový vyhodnocovač

Ukázkový vyhodnocovač načítá vstup v následujícím formátu:

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

Ukázkový vyhodnocovač vypisuje váš výstup v následujícím formátu:

- řádek 1: návratová hodnota funkce `find_maximum`
- řádek $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$