



Sumljive Srečke (tickets)

Ringo je na karnevalu v Singapurju. Pri sebi ima nekaj srečk, ki bi jih rad unovčil na stojnici z nagradami. Vsaka srečka je pobarvana z eno izmed n barv, na njej pa je neko nenegativno celo število. Posamezno število se lahko pojavi na več različnih srečkah. Zagotovljeno je, da je n **sodo** število.

Ringo ima pri sebi m srečk vsake barve, torej skupaj $n \cdot m$ srečk. Na j -ti srečki i -te barve je število $x[i][j]$ ($0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$).

Nagrade se žreba v k krogih, oštevilčenih od 0 do $k - 1$. Vsak krog poteka takole:

- Ringo izbere izmed svojih srečk **množico** n srečk, eno vsake barve. To množico nato da vodi žreba.
- Vodja žreba si zapiše števila $a[0], a[1] \dots a[n - 1]$, ki so napisane na srečkah iz množice. Vrstni red teh n števil ni pomemben.
- Vodja žreba izžreba srečno karto in si zapiše celo število b , ki je napisano na karti.
- Vodja igre za vsak i ($0 \leq i \leq n - 1$) izračuna absolutno vrednost razlike $a[i]$ in b . Naj bo S vsota teh absolutnih vrednosti.
- Za ta krog Ringo dobi nagrado v vrednosti S .
- Uporabljenih srečk se zavrže in jih ni možno uporabiti v naslednjih krogih.

Srečke, ki jih Ringo ni porabil v k krogih žreba, ostanejo neporabljene.

S pozornim opazovanjem je Ringo ugotovil, da je žreb nepošten! Vodja žreba v vsakem krogu poišče tak b , ki minimizira vrednost nagrade v tem krogu, in ta b napiše na srečno karto.

S tem znanjem bi Ringo rad razporedil svoje srečke po posameznih krogih žreba. Za vsak krog bi rad izbral tako množico srečk, da maksimizira skupno vrednost dobljenih nagrad.

Podrobnosti implementacije

Implementiraj naslednji funkciji:

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

- k : število krogov žreba.
- x : $n \times m$ polje števil na posameznih srečkah. Srečke posamezne barve so urejene v nepadajočem vrstnem redu glede na števila na njih.
- Ta funkcija se pokliče natanko enkrat.

- Funkcija naj izvede natanko en klic `allocate_tickets` z razporeditvijo k množic srečk, eno za vsak krog žreba. Razporeditev naj maksimizira skupno vrednost nagrad.
- Funkcija naj vrne skupno vrednost nagrad.

Funkcija `allocate_tickets` je definirana kot:

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

- s : $n \times m$ polje. Element $s[i][j]$ naj bo r , če je j -ta srečka i -te barve uporabljena v r -tem krogu, oziroma -1 , če sploh ni uporabljena.
- Za vsak $0 \leq i \leq n - 1$ naj se v $s[i][0], s[i][1], \dots, s[i][m - 1]$ vsako število $0, 1, 2, \dots, k - 1$ pojavi natanko enkrat, ostali elementi pa morajo biti -1 .
- Če je možnih več optimalnih rešitev, lahko vrneš katerokoli izmed njih.

Primeri

1. primer

Obravnavamo naslednji klic:

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

To pomeni:

- obstaja $k = 2$ krogov;
- števila, natisnjena na srečkah barve 0, so 0, 2 in 5;
- števila, natisnjena na srečkah barve 1 so 1, 1 in 3.

Ena izmed možnih razporeditev, ki da maksimalno nagrado je:

- V 0. krogu, Ringo izbere srečko 0 barve 0 (s številom 0) in srečko 2 barve 1 (s številom 3). Najnižja možna vrednost nagrade v tem krogu je 3. Npr. vodja žreba izbere $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- V 1. krogu, Ringo izbere srečko 2 barve 0 (s številom 5) in srečko 1 barve 1 (s številom 1). Najnižja možna vrednost nagrade v tem krogu je 4. Npr. vodja žreba izbere $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Zatorej, skupna vrednost nagrad je $3 + 4 = 7$.

Da sporočimo to rešitev, funkcija `find_maximum` izvede klic `allocate_tickets`:

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

Za tem funkcija `find_maximum` vrne 7.

2. primer

Obravavamo naslednji klic:

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

To pomeni:

- obstaja samo en krog,
- števila na srečkah barve 0 so 5 in 9;
- števila na srečkah barve 1 so 1 in 4;
- števila na srečkah barve 2 so 3 in 6;
- števila na srečkah barve 3 so 2 in 7.

Ena izmed možnih razporeditev, ki da maksimalno nagrado je:

- V krogu 0, Ringo izbere srečko 1 barve 0 (s številko 9), srečko 0 barve 1 (s številko 1), srečko 0 barve 2 (s številko 3), in srečko 1 barve 3 (s številko 7). Najnižja možna vrednost nagrade v tem krogu je 12, ko vodja žreba izbere $b = 3$:
 $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Da sporočimo to rešitev, funkcija `find_maximum` izvede klic `allocate_tickets`:

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

Za tem funkcija `find_maximum` vrne 12.

Omejitve

- $2 \leq n \leq 1500$ in n je sodo.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (za vse $0 \leq i \leq n - 1$ in $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (za vse $0 \leq i \leq n - 1$ in $1 \leq j \leq m - 1$)

Podnaloge

1. (11 točk) $m = 1$
2. (16 točk) $k = 1$
3. (14 točk) $0 \leq x[i][j] \leq 1$ (za vse $0 \leq i \leq n - 1$ in $0 \leq j \leq m - 1$)
4. (14 točk) $k = m$
5. (12 točk) $n, m \leq 80$
6. (23 točk) $n, m \leq 300$
7. (10 točk) Ni dodatnih omejitev.

Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v naslednjem formatu:

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

Vzorčni ocenjevalnik izpiše tvoj odgovor v naslednjem formatu:

- vrstica 1: vrednost, ki jo vrne `find_maximum`
- vrstica $2 + i$: ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$