

Карневалски тикети (тикети)

Благојче е на карневал во Сингапур. Во ранецот тој има неколку наградни тикети, кои би сакал да ги искористи на тезгата за наградна игра. Секој тикет доаѓа во една од n бои и на него има отпечатено еден ненегативен цел број. Може да се случи на различни тикети да е отпечатен ист број. Заради невообичаените правила на карневалот, се гарантира дека n е секогаш парен.

Благојче има по m тикети од секоја боја во неговиот ранец, односно, тоа се вкупно $n \cdot m$ тикети. На тикетот j од бојата i е отпечатен целиот број $x[i][j]$ ($0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$).

Наградната игра се игра во k круга, нумерирани со целите броеви од 0 до $k - 1$. Секој круг се игра по следниот редослед:

- Од неговиот ранец, Благојче одбира **множество** од n тикети, по еден тикет од секоја боја. Тој потоа му го дава множеството на управувачот со играта (анг. game master).
- Управувачот со играта ги запишува броевите $a[0], a[1] \dots a[n - 1]$ отпечатени на тикетите од множеството. Редоследот на овие n броеви не е важен.
- Управувачот со играта извлекува една специјална карта од кутијата на среќата (кутија со карти) и го запишува бројот b отпечатен на оваа карта.
- Управувачот со играта ги пресметува апсолутните разлики помеѓу $a[i]$ и b за секое i од 0 до $n - 1$. Нека S е збирот на овие апсолутни разлики.
- За тековниот круг, управувачот со играта на Благојче му дава награда со вредност еднаква на S .
- Тикетите од множеството се отстрануваат и не можат да се користат во идните кругови.

Преостанатите тикети во ранецот на Благојче после k -те кругови од играта се отстрануваат.

Гледајќи внимателно, Благојче сфатил дека наградната игра е наместена! Всушност, постои еден печатач во внатрешноста на кутијата на среќата. Во секој круг, управувачот со играта го наоѓа целиот број b кој што ја минимизира вредноста на наградата за тој круг. Вредноста одбрана од управувачот се печати на специјалната карта за тој круг.

Имајќи ги сите овие информации, Благојче би сакал да алоцира тикети за круговите на играта. Со други зборови, тој сака да го одбере множеството од тикети што ќе се користи во секој круг така што ќе ја максимизира вкупната вредност на наградите.

Имплементациски детали

Треба да ја имплементирате следнава процедура:

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

- k : бројот на кругови.
- x : низа со големина $n \times m$ што ги опишува броевите отпечатени на тикетите. Тикетите од секоја боја се подредени (сортирани) во неопаѓачки редослед според нивните броеви.
- Оваа процедура се повикува точно еднаш.
- Оваа процедура треба да направи точно еден повик до `allocate_tickets` (види подолу), опишувајќи k множества од тикети, по едно за секој круг. Алокацијата треба да ја максимизира вкупната вредност на наградите.
- Оваа процедура треба да ја врати максималната вкупна вредност на наградите.

Процедурата `allocate_tickets` е дефинирана на следниот начин:

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

- s : низа со големина $n \times m$. Вредноста на $s[i][j]$ треба да биде r ако тикетот j од бојата i е искористен во множеството за кругот r од играта, или пак -1 ако не е воопшто искористен.
- За секое $0 \leq i \leq n - 1$, помеѓу $s[i][0], s[i][1], \dots, s[i][m - 1]$ секоја од вредностите $0, 1, 2, \dots, k - 1$ мора да се појави точно еднаш, а сите други елементи мора да бидат -1 .
- Ако постојат повеќе алокации што доведуваат до иста максимална вкупна вредност на наградите, дозволено е да се пријави која било од нив.

Примери

Пример 1

Да го разгледаме следниот повик:

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

Ова значи дека:

- има $k = 2$ круга;
- целите броеви отпечатени на тикетите со боја 0 се 0, 2 и 5, соодветно;
- целите броеви отпечатени на тикетите со боја 1 се 1, 1 и 3, соодветно.

Можна алокација која ја дава максималната вкупна награда е:

- Во кругот 0, Благојче го одбира тикетот 0 со боја 0 (со бројот 0) и тикетот 2 со боја 1 (со бројот 3). Најмалата можна вредност на наградата во овој круг е 3. т.е., управувачот со играта може да одбере $b = 1: |1 - 0| + |1 - 3| = 1 + 2 = 3$.
- Во кругот 1, Благојче го одбира тикетот 2 со боја 0 (со бројот 5) и тикетот 1 со боја 1 (со

бројот 1). Најмалата можна вредност на наградата во овој круг е 4. т.е., управувачот со играта може да одбере $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.

- Според тоа, вкупната вредност на наградите би била $3 + 4 = 7$.

За да ја пријави оваа алокација, процедурата `find_maximum` треба да го направи следниот повик до `allocate_tickets`:

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

Конечно, процедурата `find_maximum` треба да врати 7.

Пример 2

Да го разгледаме следниот повик:

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

Ова значи дека:

- има само еден круг,
- броевите отпечатени на тикетите со боја 0 се 5 и 9, соодветно;
- броевите отпечатени на тикетите со боја 1 се 1 и 4, соодветно;
- броевите отпечатени на тикетите со боја 2 се 3 и 6, соодветно;
- броевите отпечатени на тикетите со боја 3 се 2 и 7, соодветно.

Можна алокација која ја дава максималната вкупна награда е:

- Во кругот 0, Благојче го одбира тикетот 1 со боја 0 (со бројот 9), тикетот 0 со боја 1 (со бројот 1), тикетот 0 со боја 2 (со бројот 3) и тикетот 1 со боја 3 (со бројот 7). Најмалата можна вредност на наградата во овој круг е 12, кога управувачот со играта ќе одбере $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

За да го пријави ова решение, процедурата `find_maximum` треба да го направи следниот повик до `allocate_tickets`:

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

Конечно, процедурата `find_maximum` треба да врати 12.

Ограничувања

- $2 \leq n \leq 1500$ и n е парен.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (за секое $0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (за секое $0 \leq i \leq n - 1$ и $1 \leq j \leq m - 1$)

Подзадачи

1. (11 поени) $m = 1$
2. (16 поени) $k = 1$
3. (14 поени) $0 \leq x[i][j] \leq 1$ (за секое $0 \leq i \leq n - 1$ и $0 \leq j \leq m - 1$)
4. (14 поени) $k = m$
5. (12 поени) $n, m \leq 80$
6. (23 поени) $n, m \leq 300$
7. (10 поени) Нема дополнителни ограничувања.

Пример оценувач

Пример оценувачот ги чита влезните податоци во следниот формат:

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

Пример оценувачот го печати вашиот одговор во следниот формат:

- линија 1: повратната вредност на `find_maximum`
- линии $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$