

Carnival Tickets (tickets)

Gospodin Malnar je, kako to obično biva, na IOI krenuo nešto ranije od ostatka ekipe kako bi istražio gastro ponudu, salone za masažu i vodene atrakcije, a sve to kako bi svojim olimpijcima prilikom dolaska mogao stvoriti optimalno okruženje.

Trenutno se, strogo poslovno, nalazi na jednom karnevalu u Singapuru. Kupio je neke nagradne listiće koje želi iskoristiti na jednoj od popularnih igara. Svaki listić dolazi u jednoj od n boja i na sebi ima napisan jedan nenegativan cijeli broj. Gospodin Malnar je odmah primijetio da isti broj može biti napisan na više listića te da je n **paran broj**.

Gospodin Malnar je kupio m listića svake boje, odnosno ima ukupno $n \cdot m$ listića. Listić j boje i na sebi ima napisan broj $x[i][j]$ ($0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$).

Karnevalska igra se sastoji od k rundi, numeriranih od 0 do $k - 1$. Svaka runda se igra na sljedeći način:

- Gospodin Malnar odabere **skup** od n listića, takav da je svaki listić različite boje. Taj skup listića daje šefu igre (game master).
- Šef igre promotri brojeve $a[0], a[1] \dots a[n - 1]$ napisane na listićima iz skupa kojeg je gospodin Malnar odabrao. Redoslijed tih n brojeva nije važan.
- Šef igre tada izvlači posebnu kartu koju je izvukao iz posebnog mjesta te pokazuje gospodinu Malnaru broj b koji piše na toj specijalnoj karti.
- Šef igre tada računa apsolutnu vrijednost razlike između brojeva $a[i]$ i b za svaki i između 0 i $n - 1$. Označimo sa S sumu tih apsolutnih razlika.
- Šef igre tada uručuje gospodinu Malnaru nagradu u vrijednosti od S singapurskih dolara.
- Svi listići korišteni u ovoj rundi igre se odbacuju i više se ne mogu iskoristiti u budućim rundama.

Preostali listići nakon k rundi se također odbacuju.

Gospodin Malnar, lukava lija, naše gore list, odmah je naslutio da je igra namještena! Naime, na posebnom mjestu iz kojeg šef igre izvlači svoju posebnu kartu nalazi se mali (3D isprintani) printer. Šef igre namjesti broj b tako da je vrijednost nagrade S u svakoj rundi najmanja moguća.

Naoružan ovim informacijama, gospodin Malnar vraća se još jači. Svoje će poteze odigrati na način da maksimizira ukupnu vrijednost nagrada na kraju igre.

Implementacijski detalji

Potrebno je implementirati sljedeću proceduru:

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

- k : broj rundi igre.
- x : polje dimenzija $n \times m$ koje opisuje svaki listić. Listići svake boje poredani su u neopadajućem poretku prema brojevima koji se na njima nalaze.
- Ova će procedura biti pozvana točno jednom.
- U ovoj proceduri potrebno je točno jednom pozvati proceduru `allocate_tickets` (vidi dolje), kojom je potrebno opisati svaki od k skupova listića koje će gospodin Malnar iskoristiti tokom igre. Odabir ovih k skupova mora biti takav da gospodin Malnar u konačnici osvoji nagrade čija je ukupna vrijednost maksimalna.
- Procedura treba vratiti maksimalnu ukupnu vrijednost osvojenih nagrada.

Procedura `allocate_tickets` definirana je kao:

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

- s : polje dimenzija $n \times m$. Vrijednost $s[i][j]$ treba biti jednaka r ako listić j boje i treba biti dio skupa r -te runde igre. Ako se taj listić neće uopće koristiti, vrijednost $s[i][j]$ treba biti jednaka -1 .
- Za svaki $0 \leq i \leq n - 1$, među vrijednostima $s[i][0], s[i][1], \dots, s[i][m - 1]$ svaki broj $0, 1, 2, \dots, k - 1$ mora se pojaviti točno jednom, a sve ostale vrijednosti moraju biti jednake -1 .
- Ako postoji više različitih rješenja koja rezultiraju maksimalnom ukupnom vrijednošću nagrada, dopušteno je prijaviti bilo koje.

Probni primjeri

Primjer 1

Razmotrimo sljedeći poziv:

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

Vidimo da:

- postoje $k = 2$ runde;
- brojevi napisani na listićima boje 0 su 0, 2 i 5;
- brojevi napisani na listićima boje 1 su 1, 1 i 3.

Jedan od načina za dobivanje nagrada najveće ukupne vrijednosti jest:

- U rundi 0, gospodin Malnar odabire listić 0 boje 0 (na kome je napisan broj 0) i listić 2 boje 1 (na kome je napisan broj 3). Najmanja moguća vrijednost nagrade u toj rundi je 3. Primjerice, šef igre može odabrati $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.

- U rundi 1, gospodin Malnar odabire listić 2 boje 0 (na kome je napisan broj 5) i listić 1 boje 1 (na kome je napisan broj 1). Najmanja moguća vrijednost nagrade u toj rundi je 4. Primjerice, šef igre može odabrati $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Stoga, ukupna vrijednost nagrada iznosi $3 + 4 = 7$.

Procedura `find_maximum` treba pozvati proceduru `allocate_tickets` na sljedeći način:

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

Konačno, procedura `find_maximum` treba vratiti 7.

Primjer 2

Razmotrimo sljedeći poziv:

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

Vidimo da:

- postoji samo jedna runda,
- brojevi napisani na listićima boje 0 su 5 i 9;
- brojevi napisani na listićima boje 1 su 1 i 4;
- brojevi napisani na listićima boje 2 su 3 i 6;
- brojevi napisani na listićima boje 3 su 2 i 7.

Jedan od načina za dobivanje nagrada najveće ukupne vrijednosti jest:

- U rundi 0, gospodin Malnar odabire listić 1 boje 0 (na kome je napisan broj 9), listić 0 boje 1 (na kome je napisan broj 1), listić 0 boje 2 (na kome je napisan broj 3) i listić 1 boje 3 (na kome je napisan broj 7). Najmanja moguća vrijednost nagrade u ovoj rundi je 12, kada šef igre odabire $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Procedura `find_maximum` treba pozvati proceduru `allocate_tickets` na sljedeći način:

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

Konačno, procedura `find_maximum` treba vratiti 12.

Ograničenja

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

Podzadaci

1. (11 points) $m = 1$
2. (16 points) $k = 1$
3. (14 points) $0 \leq x[i][j] \leq 1$ (za sve $0 \leq i \leq n - 1$ i $0 \leq j \leq m - 1$)
4. (14 points) $k = m$
5. (12 points) $n, m \leq 80$
6. (23 points) $n, m \leq 300$
7. (10 points) Nema dodatnih ograničenja.

Ogledni ocjenjivač

Ogledni ocjenjivač čita ulaz u sljedećem formatu:

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

Ogledni ocjenjivač ispisuje vaše rješenje u sljedećem formatu:

- redak 1: izlazna vrijednost procedure `find_maximum`
- redak $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$