



Tiket Karnaval (tickets)

Ringo sedang berada di karnaval di Singapura. Dia memiliki beberapa tiket hadiah di tasnya, yang akan dia gunakan di kios *game* berhadiah. Tiap tiket memiliki salah satu dari n warna dan terdapat sebuah bilangan bulat non-negatif yang tertera di tiket tersebut. Bilangan-bilangan yang tertera di tiket berbeda mungkin memiliki nilai yang sama. Karena suatu karakteristik pada aturan karnaval, n dijamin merupakan bilangan **genap**.

Ringo memiliki m tiket untuk setiap warna di tasnya, dengan total $n \cdot m$ tiket. Tiket j dari warna i memiliki bilangan $x[i][j]$ yang tertera di tiket tersebut ($0 \leq i \leq n - 1$ dan $0 \leq j \leq m - 1$).

Game berhadiah dimainkan dalam k ronde, dinomori dari 0 sampai $k - 1$. Tiap ronde dimainkan dengan urutan langkah sebagai berikut:

- Dari tasnya, Ringo memilih sebuah **himpunan** berisikan n buah kartu, satu kartu untuk setiap warna. Dia kemudian memberikan himpunan tersebut ke *game master*.
- *Game master* mencatat bilangan $a[0], a[1] \dots a[n - 1]$ yang tertera pada tiket di himpunan tersebut. Urutan dari n bilangan ini tidak penting.
- *Game master* menarik sebuah kartu spesial dari kotak undian dan menulis bilangan bulat b yang tertera pada kartu spesial tersebut.
- *Game master* menghitung selisih dari $a[i]$ dan b untuk setiap i dari 0 sampai $n - 1$. Didefinisikan S sebagai jumlah dari selisih-selisih tersebut.
- Untuk ronde ini, *game master* akan memberikan Ringo hadiah bernilai S
- Tiket-tiket di himpunan dibuang dan tidak dapat digunakan lagi di ronde-ronde berikutnya.

Tiket-tiket yang tersisa di tas Ringo setelah k ronde *game* akan dibuang.

Dengan memperhatikan dengan seksama, Ringo menyadari bahwa terdapat kecurangan di *game* berhadiah tersebut! Terdapat printer di dalam kotak undian. Di setiap ronde, *game master* mencari bilangan b yang meminimalkan nilai dari hadiah di rondei itu. Nilai yang dipilih *game master* akan dicetak di kartu spesial di ronde itu.

Dengan mengetahui semua informasi ini, Ringo ingin mengalokasikan tiket-tiket ke ronde-ronde dari *game*. Dengan kata lain, dia ingin memilih himpunan tiket di setiap ronde untuk memaksimalkan total nilai hadiah yang didapat.

Detail Implementasi

Anda harus mengimplementasikan fungsi berikut:

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

- k : banyaknya ronde
- x : Sebuah array berukuran $n \times m$ yang mendeskripsikan bilangan yang tertera pada setiap tiket. Tiket-tiket dengan warna yang sama diurutkan dengan urutan tidak menurun berdasarkan bilangan yang tertera pada tiket.
- Fungsi ini dipanggil tepat sekali
- Fungsi ini harus memanggil `allocate_tickets` tepat sekali (lihat detail dibawah), mendeskripsikan k himpunan tiket, satu untuk setiap ronde. Alokasi tiket harus memaksimalkan total nilai dari hadiah.
- Fungsi ini harus mengembalikan nilai maksimum dari total hadiah.

Fungsi `allocate_tickets` didefinisikan sebagai berikut:

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

- s : Sebuah array berukuran $n \times m$. Nilai dari $s[i][j]$ harus bernilai r apabila tiket j dari warna i digunakan di himpunan dari ronde r di *game*, atau -1 apabila tidak digunakan sama sekali.
- Untuk setiap $0 \leq i \leq n - 1$, dari $s[i][0], s[i][1], \dots, s[i][m - 1]$ setiap bilangan $0, 1, 2, \dots, k - 1$ harus muncul tepat sekali, dan semua entri lainnya harus bernilai -1 .
- Apabila terdapat beberapa alokasi yang menghasilkan total hadiah maksimum, keluarkan yang mana saja.

Contoh

Contoh 1

Perhatikan pemanggilan berikut:

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

Hal ini berarti:

- terdapat $k = 2$ ronde;
- bilangan yang tertera pada tiket berwarna 0 adalah 0, 2 dan 5;
- bilangan yang tertera pada tiket berwarna 1 adalah 1, 1 dan 3;

Sebuah kemungkinan alokasi yang memberikan total hadiah maksimum adalah:

- Di ronde 0, Ringo memilih tiket 0 dari warna 0 (yang tertera bilangan 0) dan tiket 2 dari warna 1 (yang tertera bilangan 3). Nilai hadiah terkecil yang mungkin pada ronde ini adalah 3, ketika *game master* memilih $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- Di ronde 1, Ringo memilih tiket 2 dari warna 0 (yang tertera bilangan 5) dan tiket 1 dari warna 1 (yang tertera bilangan 1). Nilai hadiah terkecil yang mungkin pada ronde ini adalah 4, ketika

game master memilih $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.

- Sehingga, total nilai hadiah adalah $3 + 4 = 7$.

Untuk melaporkan alokasi ini, fungsi `find_maximum` harus melakukan pemanggilan berikut ke `allocate_tickets`:

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

Akhirnya, fungsi `find_maximum` harus mengembalikan 7.

Contoh 2

Perhatikan pemanggilan berikut:

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

Hal ini berarti:

- hanya terdapat satu ronde,
- bilangan yang tertera pada tiket berwarna 0 adalah 5 dan 9;
- bilangan yang tertera pada tiket berwarna 1 adalah 1 dan 4;
- bilangan yang tertera pada tiket berwarna 2 adalah 3 dan 6;
- bilangan yang tertera pada tiket berwarna 3 adalah 2 dan 7;

Sebuah kemungkinan alokasi yang memberikan total hadiah maksimum adalah:

- Di ronde 0, Ringo memilih tiket 1 dari warna 0 (yang tertera bilangan 9), tiket 0 dari warna 1 (yang tertera bilangan 1), tiket 0 dari warna 2 (yang tertera bilangan 3), dan tiket 1 dari warna 3 (yang tertera bilangan 7). Nilai hadiah terkecil yang mungkin pada ronde ini adalah 12, ketika *game master* memilih $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Untuk melaporkan solusi ini, fungsi `find_maximum` harus melakukan pemanggilan berikut ke `allocate_tickets`:

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

Akhirnya, fungsi `find_maximum` harus mengembalikan 12.

Batasan

- $2 \leq n \leq 1500$ dan n merupakan bilangan genap.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (untuk setiap $0 \leq i \leq n - 1$ dan $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (untuk setiap $0 \leq i \leq n - 1$ dan $1 \leq j \leq m - 1$)

Subsoal

1. (11 poin) $m = 1$
2. (16 poin) $k = 1$
3. (14 poin) $0 \leq x[i][j] \leq 1$ (untuk setiap $0 \leq i \leq n - 1$ dan $0 \leq j \leq m - 1$)
4. (14 poin) $k = m$
5. (12 poin) $n, m \leq 80$
6. (23 poin) $n, m \leq 300$
7. (10 poin) Tidak ada batasan tambahan.

Contoh *grader*

Contoh *grader* akan membaca masukan dengan format berikut:

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

Contoh *grader* akan mencetak jawaban Anda dengan format berikut:

- baris 1: nilai yang dikembalikan `find_maximum`
- baris $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$