



Karnaval Biletleri (tickets)

Ringo Singapur'da düzenlenen bir karnavala katılır. Çantasında bulunan ödül biletleri ile ödül alma oyununa katılmak ister. Her bilet n farklı renkten birisindendir ve üzerinde negatif olmayan bir sayı vardır. Farklı biletlerde aynı sayılar olabilir. Karnavaldaki garip bir kural gereği, n sayısının **çift** olacağı garantidir.

Ringo'nun çantasında her renkten m tane bilet vardır, yani toplam $n \cdot m$ bilet. i rengindeki j . biletin üzerinde $x[i][j]$ sayısı vardır ($0 \leq i \leq n - 1$ ve $0 \leq j \leq m - 1$).

Ödül oyunu 0'dan $k - 1$ 'e numaralandırılmış k raundda oynanır. Her bir raund aşağıdaki belirtilen şekilde oynanır:

- İlk olarak, Ringo çantasından her bir renkten birer tane bilet olacak şekilde n büyüklüğünde bir bilet **kümesi** seçer. Sonra bu biletleri oyun yöneticisine verir.
- Sonra, oyun yöneticisi o kümedeki kartların üzerinde yazılı olan $a[0], a[1] \dots a[n - 1]$ sayılarına bakar. Bu n tane tamsayının sırası önemli değildir.
- Daha sonra, oyun yöneticisi şanslı kart kutusundan özel bir kart çeker ve o kart üzerinde yazan b tamsayısını bir yere not eder.
- Oyun yöneticisi 0'dan $n - 1$ 'e her bir i için $a[i]$ ve b arasındaki mutlak farkı hesaplar. S bütün mutlak farkların toplamı olsun.
- Bu raundda oyun yöneticisi Ringo'ya ödül olarak S değerinde bir ödül verir.
- Bu raundda kullanılan kart kümesi kenara koyulur ve sonraki raundlarda kullanılamazlar.

k raund sonunda Ringo'nun çantasında kalan kartlar kullanılmamak üzere atılır.

Ringo dikkatlice izleyince bu ödül oyununun hileli olduğunu farkeder! Şanslı kart kutusunun içinde gizli bir yazıcı vardır. Her bir raundda, oyun yöneticisi o raundda kazanılabilecek ödül değerini minimum yapacak olan b 'yi belirler. Oyun yöneticisinin belirlediği bu sayı, o raundda özel kartın üzerine yazıcı tarafından basılır.

Bütün bu bilgiler ışığında Ringo raundlarda hangi kartları kullanacağını belirlemek ister. Yani hedefi, raundlarda seçeceği bilet kümeleri ile oyunda kazanacağı ödüllerin toplam değerini maksimum yapmaktır.

Implementasyon detayları

Aşağıdaki fonksiyonu implement etmelisiniz:

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

- k : raund sayısı.
- x : $n \times m$ 'lik bir dizi. Biletler üzerindeki sayıları gösterir. Her bir renkteki biletler üzerlerindeki sayılar azalmayan şekilde sıralanmışlardır.
- Bu fonksiyon tam olarak bir kez çağrılır.
- Bu fonksiyon tam olarak bir kez `allocate_tickets` fonksiyonunu (aşağıya bakınız) çağırmalıdır ve her bir tur için bir küme olmak üzere, k tane bilet kümesini tanımlamak için kullanılır. Belirlenen kümeler toplam ödül değerini maksimize etmelidir.
- Bu fonksiyon kazanılan ödüllerin toplam değerini dönmelidir.

`allocate_tickets` fonksiyonu aşağıdaki şekilde tanımlanmıştır:

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

- s : $n \times m$ 'lik bir dizi. Eğer i rengindeki j . kart oyunun r . raundunda belirlenen kümede kullanıldıysa $s[i][j]$ değeri r olmalıdır. O kart hiçbir raundda kullanılmadıysa -1 olmalıdır.
- Her bir $0 \leq i \leq n - 1$ için, $s[i][0], s[i][1], \dots, s[i][m - 1]$ içinde $0, 1, 2, \dots, k - 1$ değerlerinden her biri tam olarak bir kez geçmelidir, ve diğer bütün değerler -1 olmalıdır.
- Eğer maksimum ödül değerine ulaşabilen birden farklı kümeleme mevcutsa herhangi birisini sonuç olarak verebilirsiniz.

Örnekler

Örnek 1

Aşağıdaki çağrıyı gözönüne alınız:

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

Bu çağrı şu anlama gelir:

- $k = 2$ tane raund vardır;
- 0 rengindeki bilet üzerindeki sayılar sırası ile 0, 2 ve 5'tir;
- 1 rengindeki bilet üzerindeki sayılar sırası ile 1, 1 ve 3'tür.

Ödül sayısını maksimize edecek olası bir kümeleme aşağıdaki gibidir:

- 0. raundda, Ringo 0 renkli 0. bileti (üzerinde 0 tamsayısı yazan) ve 1 renkli 2. bileti (üzerinde 3 tamsayısı yazan) seçer. Bu raundda kazanılabilecek en küçük ödül miktarı 3 olur. Yani ödül yöneticisi $b = 1$ seçerse: $|1 - 0| + |1 - 3| = 1 + 2 = 3$ olur.
- 1. raundda, Ringo 0 renkli 2. bileti (üzerinde 5 tamsayısı yazan) ve 1 renkli 1. bileti (üzerinde 1 tamsayısı yazan) seçer. Bu raundda kazanılabilecek en küçük ödül miktarı 4 olur. Yani ödül yöneticisi $b = 3$ seçerse: $|3 - 1| + |3 - 5| = 2 + 2 = 4$ olur.
- Yani toplam ödül miktarı $3 + 4 = 7$ olur.

Bu kümelemeyi çıktı olarak vermek için, `find_maximum` fonksiyonu `allocate_tickets`

fonksiyonunu şu şekilde çağırmalıdır:

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

Son olarak, `find_maximum` fonksiyonu değer olarak 7 dönmelidir.

Örnek 2

Aşağıdaki çağırışı gözönüne alınız:

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

Bu çağrı şu anlama gelir:

- sadece bir round vardır,
- 0 rengindeki bilet üzerindeki sayılar sırası ile 5 ve 9'dur;
- 1 rengindeki bilet üzerindeki sayılar sırası ile 1 ve 4'tür;
- 2 rengindeki bilet üzerindeki sayılar sırası ile 3 ve 6'dır;
- 3 rengindeki bilet üzerindeki sayılar sırası ile 2 ve 7'dir.

Ödül sayısını maksimize edecek olası bir kümeleme aşağıdaki gibidir:

- 0. raundda, Ringo 0 renkli 1. bileti (üzerinde 9 tamsayısı yazan), 1 renkli 0. bileti (üzerinde 1 tamsayısı yazan), 2 renkli 0. bileti (üzerinde 3 tamsayısı yazan) ve 3 renkli 1. bileti (üzerinde 7 tamsayısı yazan) seçer. Bu raundda kazanılabilecek en küçük ödül miktarı 12 olur. Yani ödül yöneticisi $b = 3$ seçerse: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$ olur.

Bu kümelemeyi çıktı olarak vermek için, `find_maximum` fonksiyonu `allocate_tickets` fonksiyonunu şu şekilde çağırmalıdır:

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

Son olarak, `find_maximum` fonksiyonu değer olarak 12 dönmelidir..

Kısıtlar

- $2 \leq n \leq 1500$ ve n çifttir.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (her bir $0 \leq i \leq n - 1$ ve $0 \leq j \leq m - 1$ için)
- $x[i][j - 1] \leq x[i][j]$ (her bir $0 \leq i \leq n - 1$ ve $1 \leq j \leq m - 1$ için)

Altgörevler

1. (11 puan) $m = 1$
2. (16 puan) $k = 1$
3. (14 puan) $0 \leq x[i][j] \leq 1$ (her bir $0 \leq i \leq n - 1$ ve $0 \leq j \leq m - 1$ için)

4. (14 puan) $k = m$
5. (12 puan) $n, m \leq 80$
6. (23 puan) $n, m \leq 300$
7. (10 puan) Ek kısıt bulunmamaktadır.

Örnek grader

Örnek grader girdiyi aşağıdaki formatta okur:

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

Örnek grader cevabınızı aşağıdaki formatta basar:

- satır 1: `find_maximum` fonksiyonunun döndüğü değer
- satır $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$