Karnaval Biletləri (tickets)

Rinqo Sinqapurda karnavaldadır. Onun çantasında hədiyyə oyun köşkündə istifadə etmək istədiyi hədiyyə biletlər var. Hər bir bilet n mümkün rəngdən bir rəngdə olur və üstündə mənfi olmayan bir tam ədəd yazılır. Müxtəlif biletlərdə yazılan ədədlər eyni ola bilər. Karnavalın qəribə qaydalarına görə, n-in **cüt** olmasına zəmanət verilir.

Rinqonun çantasında hər rəngdə m bileti, cəmi $n \cdot m$ bileti var. i rəngdə olan j biletinin üstündə $x[i][j]$ ədədi yazılıb ($0 \leq i \leq n-1$ və $0 \leq j \leq m-1$).

Hədiyyə oyunu 0dan $(k-1)$-ə qədər nömrələnmiş k raundda oynanılır. Hər bir raund bu sırada oynanılır:

- Rinqo çantasından hər rəngdən bir bilet olmaqla, n bilet seçir. Sonra o, bu kartları oyun ustadına verir.
- Oyun ustadı bu kartlarda yazılan $a[0],\ a[1]\ \ \ldots\ \ a[n-1]$ ədədlərini qeyd edir. Bu n ədədin sırası vacib deyil.
- Oyun ustadı uğurlu qutudan bir xüsusi kart çıxarır və onun üzərində yazılan b ədədini qeyd edir.
- Oyun ustadı bütün 0-dan $(n-1)$-ə qədər bütün i-lər üçün $a[i]$ və b ədədinin mütləq fərqini hesablayır. Qoy S bu mütləq fərqlərin cəmi olsun.
- Bu raund üçün oyun ustadı Rinqoya S dəyərində bir hədiyyə verir.
- İstifadə olunan biletlər atılır və gələn raundlarda istifadə edilə bilməz.

k raunddan sonra Rinqonun çantasında qalan biletlər atılır.

Diqqətlə baxaraq, Rinqo hədiyyə oyununun saxta olduğunu gördü! Əslində uğurlu qutunun içində printer var imiş. Hər raundda, oyun ustadı ele b ədədi tapır ki, həmin raund üçün hədiyyənin qiyməti minimal olsun. Sonra bu seçilmiş dəyər həmin raund üçün xüsusi kart üzərinə çap edilir.

Bütün bunları bilərək, Rinqo biletləri raundlara paylamaq istəyir. Yəni o, hər bir raund üçün elə biletləri seçmək istəyir ki, qazanacağı hədiyyələrin cəmi qiyməti maksimum olsun.

İmplementasiya Detalları

Bu proseduru icra etməlisiniz:

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

- k: Raundların sayı.
- x: Hər biletdəki ədədləri göstərən $n \times m$ ölçülü massiv. Hər rəngin biletləri üstündəki ədədlərə

görə azalmayan sırada sıralanıb.
- Bu prosedur düz bir dəfə çağrılacaq.
- Bu prosedur hər raund üçün bir ədəd olmaqla k bilet seçimini təsvir edərək, `allocate_tickets` prosedurunu (aşağıya baxın) düz bir dəfə çağırmalıdır.
- Bu prosedur hədiyyələrin cəmi qiymətlərinin maksimum dəyərini qaytarmalıdır.

`allocate_tickets` proseduru belə təyin olunur:

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

- s: $n \times m$ ölçülü massiv. Əgər i rəngində olan j bileti r raundunda istifadə edilərsə $s[i][j]$ dəyəri r-yə, yox əgər heç istifadə edilməzsə -1-ə bərabər olmalıdır.
- Bütün $0 \leq i \leq n-1$ dəyərləri üçün, $s[i][0], s[i][1], \ldots, s[i][m-1]$ dəyərləri arasında hər bir $0, 1, 2, \ldots, k-1$ dəyəri düz bir dəfə verilməlidir, və bütün başqa dəyərlər -1 olmalıdır.
- Əgər bir neçə planlama maksimum dəyərli hədiyyə ilə nəticələnirsə, bunlardan istənilən birini verə bilərsiniz.

Nümunələr

Nümunə 1

Belə bir çağrıya baxın:

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

Bu o deməkdir ki:

- $k = 2$ raund var;
- 0 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 0, 2 və 5-dir;
- 1 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 1, 1 və 3-dür.

Maksimum cəmi hədiyyə dəyəri verən mümkün planlamalardan biri:

- 0-cı raundda, Rinqo 0 rəngli 0 biletini (0 ədədi yazılan) və 1 rəngli 2 biletini (3 ədədi yazılan) seçir. Bu raund üçün, mümkün ən kiçik hədiyyə dəyəri 3-dür. Məsələn, oyun ustadı $b = 1$ seçə bilər: $|1-0| + |1-3| = 1 + 2 = 3$.
- 1-ci raundda, Rinqo 0 rəngli 2 biletni (5 ədədi yazılan) və 1 rəngli 1 biletini (1 ədədi yazılan) seçir. Bu raund üçün, mümkün ən kiçik hədiyyə dəyəri 4-dür. Məsələn, oyun ustadı $b = 3$ seçə bilər: $|3-1| + |3-5| = 2 + 2 = 4$.
- Ona görə də, hədiyyələrin cəmi qiyməti $3 + 4 = 7$-dir.

Belə bir planlamanı bildirmək üçün, `find_maximum` proseduru `allocate_tickets` proseduruna belə bir çağrı yerinə yetirməlidir:

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

Sonda, `find_maximum` proseduru 7 qaytarır.

Nümunə 2

Belə bir çağrıya baxın:

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

Bu o deməkdir ki:

- Yalnız 1 raund var,
- 0 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 5 və 9-dur;
- 1 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 1 və 4-dür;
- 2 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 3 və 6-dır;
- 3 rəngli biletlərdə yazəlan ədədlər, uyğun olaraq, 2 və 7-dir.

Maksimum cəmi hədiyyə dəyəri verən mümkün planlamalardan biri:

- 0-cı raundda, Rinqo 0 rəngli 1 biletini (9 ədədi yazılan), 1 rəngli 0 biletini (1 ədədi yazılan), 2 rəngli 0 biletini (3 ədədi yazılan), və 3 rəngli 1 biletini (7 ədədi yazılan) seçir. Bu raund üçün, mümkün ən kiçik hədiyyə dəyəri 12-dir. Məsələn, oyun ustadı $b = 3$ seçə bilər: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Belə bir planlamanı bildirmək üçün, `find_maximum` proseduru `allocate_tickets` proseduruna belə bir çağrı yerinə yetirməlidir:

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

Sonda, `find_maximum` proseduru 12 qaytarır.

Limitlər

- $2 \leq n \leq 1500$ və n cüt ədəddir.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (bütün $0 \leq i \leq n - 1$ və $0 \leq j \leq m - 1$ üçün)
- $x[i][j - 1] \leq x[i][j]$ (bütün $0 \leq i \leq n - 1$ və $1 \leq j \leq m - 1$ üçün)

Alt tapşırıqlar

1. (11 points) $m = 1$
2. (16 points) $k = 1$
3. (14 points) $0 \leq x[i][j] \leq 1$ (bütün $0 \leq i \leq n - 1$ və $0 \leq j \leq m - 1$ üçün)
4. (14 points) $k = m$
5. (12 points) $n, m \leq 80$
6. (23 points) $n, m \leq 300$
7. (10 points) Əlavə limit yoxdur.

Nümunə Grader (Qiymətləndirici)

Nümunə grader girişi bu formatda alır:

- sətir 1: $n \;\; m \;\; k$
- sətir $2 + i$ ($0 \le i \le n - 1$): $x[i][0] \;\; x[i][1] \;\; \ldots \;\; x[i][m-1]$

Nümunə grader bu formatda sizin cavabınızı çıxışa verir:

- sətir 1: `find_maximum`-un qaytardığı dəyər
- sətir $2 + i$ ($0 \le i \le n - 1$): $s[i][0] \;\; s[i][1] \;\; \ldots \;\; s[i][m-1]$