კარნავალის ბილეთები (tickets)

რინგო კარნავალზეა სინგაპურში. მას ჩანთაში აქვს საპრიზო ბილეთები, რომელთა გამოყენებაც სურს ჯიხურში საპრიზო გათამაშებაში. თითოეულ ბილეთს აქვს n ფერიდან ერთ-ერთი და აწერია მთელი არაუარყოფითი რიცხვი. სხვადასხვა ბილეთებზე შესაძლოა ერთი და იგივე რიცხვები ეწეროს. კარნავალის წესებიდან გამომდინარე გარანტირებულია, რომ n არის **ლუწი**.

რინგოს ჩანთაში აქვს თითოეული ფერის m ცალი ბილეთი, ანუ სულ $n \cdot m$ რაოდენობის ბილეთი. i ფერის მქონე j-ურ ბილეთს აწერია მთელი $x[i][j]$ რიცხვი ($0 \le i \le n-1$ და $0 \le j \le m-1$).

საპრიზო გათამაშება შედგება k რაუნდისგან, რომლებიც დანომრილია 0-დან $(k-1)$-მდე. თითოეული რაუნდი თამაშდება შემდეგნაირად:

- რინგო ჩანთითან ირჩევს n ბილეთიან **სიმრავლეს**, თითო ბილეთით ყოველი ფერისთვის. ის შემდეგ ამ სიმრავლეს გადასცემს თამაშის ოსტატს.
- ოსტატი იწერს $a[0]$, $a[1]$ $\ldots$ $a[n-1]$ მთელ რიცხვებს, რომლებიც წერია სიმრავლის ბილეთებზე. ამ n რიცხვის თანმიმდევრობას მნიშვნელობა არ აქვს.
- ოსტატი თავისი იღბლიანი ყუთიდან იღებს სპეციალურ ბარათს და ნახულობს ამ ბარათზე დაწერილ მთელ b რიცხვს.
- ოსტატი ითვლის $a[i]$ და b-ს შორის აბსოლუტურ სხვაობას ყოველი i-თვის 0-დან ($n-1$)-მდე. ვთქვათ S არის ამ აბსოლუტური სხვაობების ჯამი.
- ამ რაუნდში ოსტატი რინგოს აძლევს S ღირებულების პრიზს.
- სიმრავლის ბარათები იყრება და შემდგომ რაუნდებში აღარ გამოიყენება.

k რაუნდის შემდეგ რინგოს ჩანთაში დარჩენილი ბილეთებიც გადაიყრება.

დაკვირვების შედეგად რინგომ შენიშნა, რომ საპრიზო თამაში ჩაწყობილია! იღბლიან ყუთში მოთავსებულია პრინტერი. ყოველ რაუნდში ოსტატი ირჩევს ისეთ b რიცხვს, რომელიც რაუნდის პრიზის მინიმიზაციას ახდენს. არჩეული რიცხვი შემდეგ იბეჭდება ამ რაუნდის ბარათზე.

ამ ინფორმაციის გამოყენებით რინგოს სურს გადაანაწილოს ბილეთები თამაშის რაუნდებზე. ანუ, მას სურს იპოვოს ბილეთების სიმრავლე თითოეული რაუნდისთვის ისე, რომ პრიზების ჯამური ღირებულება იყოს მაქსიმალური.

იმპლემენტაციის დეტალები

თქვენ გევალებათ შემდეგი პროცედურის იმპლემენტაცია:

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

- k: რაუნდების რაოდენობა.
- x: $n \times m$ მასივი, რომელიც აღწერს ბილეთებზე დაწერილ რიცხვებს. თითოეული ფერისთვის ბილეთები დალაგებულია რიცხვების არაკლებადი მიმდევრობით.
- ამ პროცედურის გამოიძახება ხდება ზუსტად ერთხელ.
- ამ პროცედურამ უნდა გამოიძახოს `allocate_tickets` (იხილეთ ქვემოთ) ზუსტად ერთხელ k რაოდენობის სიმრავლის აღწერით, თითოსი ყოველი რაუნდისთვის. განაწილება უნდა ახდენდეს პრიზების საერთო ღირებულების მაქსიმიზაციას.
- ამ პროცედურამ უნდა დააბრუნოს პრიზების საერთო ღირებულების მაქსიმუმი.

პროცედურა `allocate_tickets` განისაზღვრება შემდეგნაირად:

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

- s: $n \times m$ მასივი. $s[i][j]$ უნდა იყოს r, თუ i ფერის მქონე j-ური ბილეთი გამოიყენება r ნომრის მქონე რაუნდის სიმრავლეში, ან -1, თუ ის საერთოდ არ გამოიყენება.
- ყოველი $0 \leq i \leq n-1$-თვის, $s[i][0], s[i][1], \ldots, s[i][m-1]$ რიცხვებს შორის $0, 1, 2, \ldots, k-1$ მნიშვნელობები უნდა გვხვდებოდეს ზუსტად ერთხელ და სხვა მნიშვნელობები უნდა იყოს -1-ის ტოლი.
- თუ პრიზების საერთო ღირებულის მაქსიმუმი მიიღწევა ერთზე მეტი განაწილებით, მაშინ დაშვებულია ნებისმიერი მათგანის გადაცემა.

მაგალითები

მაგალითი 1

განვიხილოთ შემდეგი გამოძახება:

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

ეს ნიშნავს, რომ:

- არის $k = 2$ რაუნდი;
- 0 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 0, 2 და 5, შესაბამისად;
- 1 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 1, 1 და 3, შესაბამისად.

შესაძლო განაწილება, რომელიც იძლევა პრიზების საერთო ღირებულების მაქსიმუმს შემდეგია:

- რაუნდში 0 რინგო ირჩევს ბილეთს 0, ფერით 0 (რიცხვით 0) და ბილეთს 2, ფერით 1 (რიცხვით 3). პრიზის მინიმალური შესაძლო ღირებულებაა 3, რადგან ოსტატს შეუძლია აირჩიოს $b = 1$: $|1-0| + |1-3| = 1 + 2 = 3$.

- რაუნდში 1 რინგო ირჩევს ბილეთს 2, ფერით 0 (რიცხვით 5) და ბილეთს 1, ფერით 1 (რიცხვით 1). პრიზის მინიმალური შესაძლო ღირებულებაა 4, რადგან ოსტატს შეუძლია აირჩიოს $b = 3$: $|3-1| + |3-5| = 2 + 2 = 4$.
- შესაბამისად, პრიზების საერთო ღირებულებაა $3 + 4 = 7$.

ამ განაწილების გადასაცემად პროცედურამ `find_maximum` უნდა გამოიძახოს `allocate_tickets` შემდეგნაირად:

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

ბოლოს პროცედურამ `find_maximum` უნდა დააბრუნოს 7.

მაგალითი 2

განვიხილოთ შემდეგი გამოძახება:

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

ეს ნიშნავს, რომ:

- არის მხოლოდ ერთი რაუნდი;
- 0 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 5 და 9, შესაბამისად;
- 1 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 1 და 4, შესაბამისად;
- 2 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 3 და 6, შესაბამისად;
- 3 ფერის მქონე ბილეთებზე დაწერილი რიცხვებია 2 და 7, შესაბამისად.

შესაძლო განაწილება, რომელიც იძლევა პრიზების საერთო ღირებულების მაქსიმუმს შემდეგია:

- რაუნდში 0 რინგო ირჩევს ბილეთს 1, ფერით 0 (რიცხვით 9), ბილეთს 0, ფერით 1 (რიცხვით 1), ბილეთს 0, ფერით 2 (რიცხვით 3) და ბილეთს 1, ფერით 3 (რიცხვით 7). პრიზის მინიმალური შესაძლო ღირებულებაა 12, როდესაც ოსტატი ირჩევს $b = 3$-ს: $|3-9| + |3-1| + |3-3| + |3-7| = 6 + 2 + 0 + 4 = 12$.

ამ განაწილების გადასაცემად პროცედურამ `find_maximum` უნდა გამოიძახოს `allocate_tickets` შემდეგნაირად:

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

ბოლოს პროცედურამ `find_maximum` უნდა დააბრუნოს 12.

შეზღუდვები

- $2 \le n \le 1500$ და n ლუწია.
- $1 \le k \le m \le 1500$
- $0 \le x[i][j] \le 10^9$ ($0 \le i \le n-1$ და $0 \le j \le m-1$)

- $x[i][j-1] \leq x[i][j]$ ($0 \leq i \leq n-1$ და $1 \leq j \leq m-1$)

ქვეამოცანები

1. (11 ქულა) $m = 1$
2. (16 ქულა) $k = 1$
3. (14 ქულა) $0 \leq x[i][j] \leq 1$ ($0 \leq i \leq n-1$ და $0 \leq j \leq m-1$)
4. (14 ქულა) $k = m$
5. (12 ქულა) $n, m \leq 80$
6. (23 ქულა) $n, m \leq 300$
7. (10 ქულა) დამატებითი შეზღუდვების გარეშე.

სანიმუშო გრადერი

სანიმუშო გრადერს შეაქვს მონაცემები შემდეგი ფორმატით:

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

სანიმუშო გრადერს გამოაქვს თქვენი პასუხი შემდეგი ფორმატით:

- სტრიქონი 1: მნიშვნელობა, რომელსაც აბრუნებს `find_maximum`
- სტრიქონი $2+i$ ($0 \leq i \leq n-1$): $\ s[i][0]\ \ s[i][1]\ \ldots\ s[i][m-1]$