



Կառնավալի տոմսերը (tickets)

Ռինգոն գտնվում է Սինգապուրի կառնավալում: Նա իր պայուսակում ունի որոշ մրցանակային տոմսեր, որոնք նախընտրում է օգտագործել մրցանակային խաղի կրպակում: Յուրաքանչյուր տոմս ունի n գույներից որևէ մեկը և ունի ոչ բացասական ամբողջ թիվ տպված իր վրա: Ամբողջ թվերը՝ տպված տարբեր տոմսերի վրա կարող են լինել նույնը: Կառնավալի կանոնների յուրօրինակության պատճառներով երաշխավորվում է, որ n -ը **գույգ** է:

Ռինգոն իր պայուսակում ունի յուրաքանչյուր գույնի m տոմս, այն է գումարային $n \cdot m$ տոմս: i -րդ գույնի j -րդ տոմսի վրա տպված է $x[i][j]$ ամբողջ թիվը ($0 \leq i \leq n - 1$ և $0 \leq j \leq m - 1$).

Մրցանակային խաղը անցկացվում է k փուլերով, համարակալված 0 -ից մինչև $k - 1$: Յուրաքանչյուր փուլ անցկացվում է հետևյալ հերթականությամբ.

- Ռինգոն իր պայուսակից ընտրում է n տոմսերի **բազմություն**, մեկ տոմս ամեն գույնից: Հետո նա տալիս է այդ բազմությունը խաղավարին:
- Խաղավարը նշում է տոմսերի բազմության վրա տպված ամբողջ թվերը $a[0], a[1] \dots a[n - 1]$: Այս n ամբողջ թվերի հերթականությունը կարևոր չէ:
- Խաղավարը դուրս է քաշում յուրօրինակ խաղաքարտ հաջողակ վիճակահանության արկղից և նշում է խաղաքարտի վրա տպված b ամբողջ թիվը:
- Խաղավարը հաշվում է $a[i]$ -երի և b -ի միջև բացարձակ տարբերությունները յուրաքանչյուր i -ի համար 0 -ից մինչև $n - 1$: S -ով նշանակենք բացարձակ տարբերությունների գումարը:
- Այս փուլի համար, խաղավարը տալիս է Ռինգոյին S արժեքով մրցանակ:
- Բազմության տոմսերը դեն են նետվում և չեն կարող օգտագործվել ապագա փուլերում:

Խաղի k փուլերից հետո Ռինգոն պայուսակում մնացած տոմսերը դեն են նետվում:

Ուշադիր նայելով Ռինգոն հասկացավ, որ մրցանակային խաղը կեղծված է! Իրականում հաջողակ վիճակահանության արկղի ներսում կա տպիչ: Յուրաքանչյուր փուլում, խաղավարը գտնում է b ամբողջ թիվ, որը մինիմիզացնում է այդ փուլի մրցանակի արժեքը: Խաղավարի գտած արժեքը տպվում է յուրօրինակ խաղաքարտի վրա այդ փուլի համար:

Ունենալով բոլոր այս տեղեկությունները, Ռինգոն ցանկանում է առանձնացնել խաղի փուլերի տոմսերը: Նա ցանկանում է ընտրել յուրաքանչյուր փուլում օգտագործվող այն տոմսերի բազմությունները, որոնք կմաքսիմիզացնեն

գումարային մրցանակների արժեքները:

Իրականացման մանրամասներ

Դուք պետք է իրականացնեք հետևյալ ֆունկցիաները.

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

- k . փուլերի քանակը:
- x . $n \times m$ մատրից նկարագրող յուրաքանչյուր տոմսի վրայի ամբողջ թվերը: Յուրաքանչյուր գույնի տոմսերը սորտավորված են նրանց ամբողջ թվերի չնվազման կարգով:
- Այս ֆունկցիան կանչվում է ճիշտ մեկ անգամ:
- Այս ֆունկցիան պետք է կատարի `allocate_tickets`-ի (տեսեք ներքևում) ճիշտ մեկ կանչ, նկարագրող k տոմսերի բազմությունները, մեկը ամեն փուլի համար. Առանձնացումը պետք է մաքսիմիզացնի մրցանակների արժեքների գումարը:
- Այս ֆունկցիան պետք է վերադարձնի առավելագույն մրցանակների արժեքների գումարը:

`allocate_tickets` ֆունկցիան սահմանվում է հետևյալ կերպ.

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

- s . $n \times m$ մատրից: $s[i][j]$ -ի արժեքը պետք է լինի r , եթե i -րդ գույնի j -րդ տոմսը օգտագործվում է խաղի r -րդ փուլի բազմությունում, կամ -1 եթե դա չի օգտագործվում ընդհանրապես:
- Յուրաքանչյուր $0 \leq i \leq n - 1$ համար, $s[i][0], s[i][1], \dots, s[i][m - 1]$ -ների միջև $0, 1, 2, \dots, k - 1$ յուրաքանչյուր արժեք պետք է հանդիպի ճիշտ մեկ անգամ, և բոլոր մնացածները պետք է լինեն -1 -եր:
- Եթե կան բազմակի առանձնացումներ՝ առավելագույն մրցանակների արժեքների գումարին բերող, թույլատրվում է հաղորդել նրանցից որևէ մեկը:

Օրինակներ

Օրինակ 1

Ենթադրենք ունենք հետևյալ կանչը.

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

Դա նշանակում է որ.

- կան $k = 2$ փուլեր;
- ամբողջ թվերը տպված 0-րդ գույնի տոմսերի վրա 0-ն, 2-ն և 9-ն են,

համապատասխանաբար;

- ամբողջ թվերը տպված 1-ին գույնի տոմսերի վրա 1-ն, 1-ն և 3-ն են, համապատասխանաբար;

Հնարավոր առանձնացումը, որը տալիս է մաքսիմալ գումարային մրցանակային արժեք.

- 0-րդ փուլում, Ռինգոն ընտրում է 0-րդ գույնի 0 ինդեքսով տոմսը (0 ամբողջ արժեքով) և 1-ին գույնի 2 ինդեքսով տոմսը (3 ամբողջ արժեքով): Մրցանակի նվազագույն հնարավոր արժեքը այս փուլում 3 է: Օրինակ, երբ խաղի վարպետը ընտրի $b = 1$. $|1 - 0| + |1 - 3| = 1 + 2 = 3$:
- 1-ին փուլում, Ռինգոն ընտրում է 0-րդ գույնի 2 ինդեքսով տոմսը (5 ամբողջ արժեքով) և 1-ին գույնի 1 ինդեքսով տոմսը (1 ամբողջ արժեքով). Մրցանակի նվազագույն հնարավոր արժեքը այս փուլում 4 է: Օրինակ, երբ խաղավարն ընտրի $b = 3$. $|3 - 1| + |3 - 5| = 2 + 2 = 4$:
- Այսպիսով, մրցանակների գումարային արժեքը կլինի $3 + 4 = 7$:

Այս լուծումը հաղորդելու համար, `find_maximum` ֆունկցիան պետք է կատարի `allocate_tickets`-ի հետևյալ կանչը.

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

Վերջնական, `find_maximum` ֆունկցիան պետք է վերադարձնի 7:

Օրինակ 2

Ենթադրենք ունենք հետևյալ կանչը.

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

Դա նշանակում է որ.

- կա ճիշտ մեկ փուլ,
- ամբողջ թվերը տպված 0-րդ գույնի տոմսերի վրա 5-ն և 9-ն են, համապատասխանաբար;
- ամբողջ թվերը տպված 1-ին գույնի տոմսերի վրա 1-ն և 4-ն են, համապատասխանաբար;
- ամբողջ թվերը տպված 2-րդ գույնի տոմսերի վրա 3-ն և 6-ն են, համապատասխանաբար;
- ամբողջ թվերը տպված 3-րդ գույնի տոմսերի վրա 2-ն և 7-ն են, համապատասխանաբար;

Հնարավոր առանձնացումը, որը տալիս է մաքսիմալ գումարային մրցանակային արժեք.

- 0-րդ փուլում, Ռինգոն ընտրում է 0-րդ գույնի 1 ինդեքսով տոմսը (9 ամբողջ արժեքով), 1-ին գույնի 0 ինդեքսով տոմսը (1 ամբողջ արժեքով), 2-րդ գույնի 0

ինդեքսով տոմսը (3 ամբողջ արժեքով), և 3-րդ գույնի 1-ին ինդեքսով տոմսը (7 ամբողջ արժեքով): Մրցանակի նվազագույն հնարավոր արժեքը այս փուլում 12 է, երբ խաղի վարպետը ընտրի $b = 3$.
 $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$:

Այս լուծումը հաղորդելու համար, `find_maximum` ֆունկցիան պետք է կատարի `allocate_tickets`-ի հետևյալ կանչը.

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

Վերջնական, `find_maximum` ֆունկցիան պետք է վերադարձնի 12:

Սահմանափակումներ

- $2 \leq n \leq 1500$ և n -ը գույգ է:
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (բոլոր $0 \leq i \leq n - 1$ և $0 \leq j \leq m - 1$ համար)
- $x[i][j - 1] \leq x[i][j]$ (բոլոր $0 \leq i \leq n - 1$ և $1 \leq j \leq m - 1$ համար)

Ենթախնդիրներ

1. (11 միավոր) $m = 1$
2. (16 միավոր) $k = 1$
3. (14 միավոր) $0 \leq x[i][j] \leq 1$ (բոլոր $0 \leq i \leq n - 1$ and $0 \leq j \leq m - 1$ համար)
4. (14 միավոր) $k = m$
5. (12 միավոր) $n, m \leq 80$
6. (23 միավոր) $n, m \leq 300$
7. (10 միավոր) Չկան հավելյալ սահմանափակումներ:

Գնահատող ծրագրի նմուշ

Գնահատող ծրագիրը կարդում է մուտքային տվյալները հետևյալ ձևաչափով.

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

Գնահատող ծրագիրը տպում է ձեր պատասխանը հետևյալ ձևաչափով.

- տող 1: վերադարձնում է `find_maximum`-ի արժեքը
- տող $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$