



Carnival Tickets (tickets)

Ringo is op een kermis in Singapore. Hij heeft een aantal prijstickets in zijn tas die hij wil gebruiken bij de prijs spel-kraam. Elk ticket heeft één van n kleuren en heeft een niet-negatief integer erop gedrukt. De integers op verschillende tickets kunnen hetzelfde zijn. Door een gekke regel van de kermis is het gegarandeerd dat n **even** is.

Ringo heeft m tickets van elke kleur in zijn tas, dat zijn in totaal $n \cdot m$ tickets. De ticket j van de kleur i heeft het integer $x[i][j]$ erop afgedrukt ($0 \leq i \leq n - 1$ en $0 \leq j \leq m - 1$)

Het prijs spel wordt gespeeld in k rondes, genummerd van 0 tot en met $k - 1$. Elke ronde wordt in de volgende volgorde gespeeld:

- Ringo selecteert uit de tickets in zijn tas een **verzameling** van n tickets, één ticket van elke kleur. Dan geeft hij deze aan de spelleider.
- De spelleider schrijft de integers $a[0], a[1] \dots a[n - 1]$ die op de tickets zijn gedrukt op. De volgorde van deze n integers is niet belangrijk.
- De spelleider trekt een speciale kaart uit een geluksdoos en schrijft de integer b die op deze kaart staat op.
- De spelleider berekent de absolute verschillen tussen $a[i]$ en b voor elke i van 0 tot en met $n - 1$. Noem S de som van deze absolute verschillen.
- Voor deze ronde geeft de spelleider Ringo een prijs met waarde gelijk aan S .
- De tickets in deze set worden dan weggegooid en kunnen niet in volgende rondes gebruikt worden.

De tickets die na k rondes overblijven in Ringo's tas worden ook weggegooid.

Door goed op te letten is Ringo erachter gekomen dat het prijs spel niet eerlijk is! Er zit namelijk een printer in de geluksdoos. In elke ronde zoekt de spelleider een integer b die de waarde van de prijs voor die ronde minimaliseert. Dit getal wordt dan op de speciale kaart van die ronde gedrukt.

Met deze informatie wil Ringo de tickets verdelen over de rondes van het spel. Dat houdt in dat hij de tickets voor elke ronde selecteert zodat de totale waarde van de prijzen gemaximaliseerd wordt.

Implementatiedetails

Je moet de volgende functie implementeren:

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

- k : het aantal rondes.

- x : een $n \times m$ array die de integers op elk ticket beschrijft. Tickets van elke kleur zijn gesorteerd in niet-dalende volgorde van hun getallen.
- Deze functie wordt één keer aangeroepen.
- Deze functie moet één keer de functie `allocate_tickets` (zie hieronder) aanroepen met een omschrijving van de k ticket sets, één voor elke ronde. De verdeling moet de totale waarde van de prijzen maximaliseren.
- De procedure moet de maximale totale waarde van de prijzen teruggeven

De functie `allocate_tickets` is als volgt gedefinieerd:

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

- s : een $n \times m$ array. De waarde van $s[i][j]$ moet r zijn als ticket j van kleur i wordt gebruikt in ronde r van het spel, of -1 als het helemaal niet gebruikt wordt.
- Voor elke $0 \leq i \leq n-1$, moet in $s[i][0], s[i][1], \dots, s[i][m-1]$ elke waarde van $0, 1, 2, \dots, k-1$ precies één keer voorkomen, alle andere waarden moeten -1 zijn.
- Als er meerdere verdelingen zijn met de maximale totale waarde van de prijzen, mag je elke daarvan opgeven.

Voorbeelden

Voorbeeld 1

Neem de volgende aanroep:

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

Dit betekent dat:

- Er $k = 2$ rondes zijn;
- De integers afgedrukt op tickets met kleur 0 zijn 0, 2 en 5 respectievelijk;
- De integers afgedrukt op tickets met kleur 1 zijn 1, 1 en 3 respectievelijk.

Een mogelijke verdeling die de maximale totale waarde van de prijzen geeft is:

- In ronde 0 kiest Ringo ticket 0 van kleur 0 (met integer 0) en ticket 2 van kleur 1 (met integer 3). De laagst mogelijke waarde van de prijs voor deze ronde is 3. De spelleider kan bijvoorbeeld kiezen $b = 1$: $|1 - 0| + |1 - 3| = 1 + 2 = 3$.
- In ronde 1 kiest Ringo ticket 2 van kleur 0 (met integer 5) en ticket 1 van kleur 1 (met integer 1). De laagst mogelijke waarde van de prijs voor deze ronde is 4. De spelleider kan bijvoorbeeld kiezen $b = 3$: $|3 - 1| + |3 - 5| = 2 + 2 = 4$.
- Daarom is de totale waarde van de prijzen $3 + 4 = 7$.

Om deze verdeling in te sturen moet de functie `find_maximum` de volgende aanroep maken:

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

Tenslotte moet de functie `find_maximum` de waarde 7 teruggeven.

Voorbeeld 2

Neem de volgende aanroep:

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

Dit betekent dat:

- Er maar één ronde is,
- De integers afgedrukt op tickets met kleur 0 zijn 5 en 9 respectievelijk;
- De integers afgedrukt op tickets met kleur 1 zijn 1 en 4 respectievelijk;
- De integers afgedrukt op tickets met kleur 2 zijn 3 en 6 respectievelijk;
- De integers afgedrukt op tickets met kleur 3 zijn 2 en 7 respectievelijk.

Een mogelijke verdeling die de maximale totale waarde van de prijzen geeft is:

- In ronde 0 kiest Ringo ticket 1 van kleur 0 (met integer 9), ticket 0 van kleur 1 (met het integer 1), ticket 0 van kleur 2 (met het integer 3) en ticket 1 van kleur 3 (met integer 7). De laagst mogelijke waarde van de prijs voor deze ronde is 12. De spelleider kan bijvoorbeeld kiezen $b = 3$: $|3 - 9| + |3 - 1| + |3 - 3| + |3 - 7| = 6 + 2 + 0 + 4 = 12$.

Om deze verdeling in te sturen moet de functie `find_maximum` de volgende aanroep maken:

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

Tenslotte moet de functie `find_maximum` de waarde 12 teruggeven.

Randvoorwaarden

- $2 \leq n \leq 1500$ en n is even.
- $1 \leq k \leq m \leq 1500$
- $0 \leq x[i][j] \leq 10^9$ (voor alle $0 \leq i \leq n - 1$ en $0 \leq j \leq m - 1$)
- $x[i][j - 1] \leq x[i][j]$ (voor alle $0 \leq i \leq n - 1$ en $1 \leq j \leq m - 1$)

Subtaken

1. (11 punten) $m = 1$
2. (16 punten) $k = 1$
3. (14 punten) $0 \leq x[i][j] \leq 1$ (voor alle $0 \leq i \leq n - 1$ en $0 \leq j \leq m - 1$)
4. (14 punten) $k = m$
5. (12 punten) $n, m \leq 80$
6. (23 punten) $n, m \leq 300$

7. (10 punten) Geen aanvullende randvoorwaarden.

Voorbeeldgrader

De voorbeeldgrader leest de invoer in het volgende formaat:

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

De uitvoer van de voorbeeldgrader is in het volgende formaat:

- regel 1: de teruggegeven waarde van `find_maximum`
- regel $2 + i$ ($0 \leq i \leq n - 1$): $s[i][0] \ s[i][1] \ \dots \ s[i][m - 1]$