# ამოცანა ბადე

⏳0.3 sec. 💾 256 MB

სიმონა ოცნებობს უზღვავ სიმდიდრეზე. მან მიიღო შეთავაზება ითამაშოს თამაში დიდი პრიზისათვის.

სიმონა თამაშს იწყებს დადებითი რიცხვებით შევსებული $N \times M$ ზომის $A$ ბადის $(0, 0)$ უჯრაზე. მან თამაში უნდა დაასრულოს $(N-1, M-1)$ უჯრაში. ამის მისაღწევად მას შეუძლია გადავიდეს მისი ახლანდელი $(x, y)$ უჯრიდან ნებისმიერ ისეთ $(x+d, y)$ ან $(x, y+d)$ უჯრაში, სადაც $d > 0$. ამ მოძრაობას გააკეთებს რამდენჯერაც უნდა და თითოეული მოძრაობისათვის იგი საჩუქრად მიიღებს $|A_{x,y} - A_{x',y'}| - C$ ცალ მონეტას, სადაც $x', y'$ მისი ახალი კოორდინატებია, ხოლო $C$ თამაშის დაწყებამდე დაფიქსირებული კონსტანტაა. შევნიშნოთ, რომ თუ $|A_{x,y} - A_{x',y'}| - C$ უარყოფითია, მაშინ სიმონას მონეტების რაოდენობა მოიკლებს. ასევე შევნიშნოთ, რომ შესაძლებელია თამაშის უარყოფითი რაოდენობის მოტენებით დასრულება.

დაეხმარეთ სიმონას დაადგინოს მონეტების მაქსიმალური რაოდენობა, რომლითაც შეუძლია თამაშის დასრულება.

*შევნიშნოთ, რომ $|a| = a$ თუ $a \geq 0$ და $|a| = -a$, სხვა შემთხვევაში.*

## იმპლემენტაციის დეტალები

თქვენ იმპლემენტაცია უნდა გაუკეთოთ შემდეგ ფუნქციას `max_profit`:

```
long long max_profit(int N, int M, int C,
std::vector<std::vector<int>> A)
```

- $N$, $M$: ბადის ზომები;
- $C$: ტესტისთვის ფიქსირებული მუდმივა;
- $A$: $N \times M$ ზომის მთელი რიცხვების ვექტორების ვექტორი, რომელიც წარმოადგენს ორგანზომილებიან ბადეს (ინდექსირებულია სტრიქონის და შემდეგ სვეტის მიხედვით).

ეს ფუნქცია გამოიძახება თითოეული ტესტისთვის ერთხელ და უნდა დააბრუნოს მონეტების მაქსიმალური რაოდენობა, რომლითაც სიმონას შეუძლია თამაშის დასრულება.

## შეზღუდვები

- $1 \leq N, M$
- $N \cdot M \leq 500\ 000$
- $1 \leq A_{i,j} \leq 1\ 000\ 000$, როცა $0 \leq i < N$ და $0 \leq j < M$
- $0 \leq C \leq 1\ 000\ 000$

## ქვეამოცანები

| ქვეამოცანა | ქულა | საჭირო ქვეამოცანები | დამატებითი შეზღუდვები |
|---|---|---|---|
| 0 | 0 | – | მაგალითი. |
| 1 | 9 | – | $N = 1, M \leq 200$ |
| 2 | 5 | – | $N = 1, A_{i,j} \leq A_{i,j+1}$ |
| 3 | 8 | – | $N = 1, C = 0$ |
| 4 | 10 | 1 | $N = 1, M \leq 50\ 000$ |
| 5 | 7 | $1 - 4$ | $N = 1$ |
| 6 | 15 | 1 | $N, M \leq 200$ |
| 7 | 9 | 2 | $A_{i,j} \leq A_{i+1,j}, A_{i,j+1}$ |
| 8 | 12 | 3 | $C = 0$ |
| 9 | 12 | $0 - 1, 4, 6$ | $N \cdot M \leq 50\ 000$ |
| 10 | 13 | $0 - 9$ | – |

## მაგალითი

განვიხილოთ შემდეგი გამოძახება:

```
max_profit(5, 6, 4, {{20, 24, 31, 33, 36, 40},
{25, 23, 25, 31, 32, 39},
{31, 26, 21, 24, 31, 35},
{32, 28, 25, 21, 26, 28},
{36, 35, 28, 24, 21, 27}})
```

ამ შემთხვევაში ოპტიმალური გზაა $(0,0) \xrightarrow{7} (0,2) \xrightarrow{2} (1,2) \xrightarrow{10} (1,5) \xrightarrow{8} (4,5)$ და მისი გაყოლით მიღებული მონეტების რაოდენობაა $7 + 2 + 10 + 8 = 27$. თქვენმა ფუნქციამ უნდა დააბრუნოს 27.

```
max_profit(2, 2, 100, {{1, 2}, {3, 4}})
```

აქ თქვენმა ფუნქციამ უნდა დააბრუნოს: $-197$. გაითვალისწინეთ, რომ პასუხის მნიშვნელობა შეიძლება იყოს უარყოფითი .

## სანიმუშო გრეიდერი

შეყვანის ფორმატი შემდეგია:

- სტრიქონი 1: სამი მთელი რიცხვი – $N$, $M$ და $C$.
- სტრიქონები $2-(N+1)$: $M$ ცალი მთელი რიცხვი – $A_{i,j}$.

გამომავალი ფორმატი შემდეგია:

- სტრიქონი 1: ერთი მთელი რიცხვი – გამოძახების მიერ დაბრუნებული მნიშვნელობა.