



Task Grid

0.3 sec. 256 MB

Simona visează la bogății nemăsurate. I se oferă să joace un joc pentru un mare premiu.

Simona este situată în celula $(0, 0)$ a unei matrice A de dimensiuni $N \times M$ ale cărei elemente sunt numere întregi pozitive. Ea trebuie să ajungă în celula $(N - 1, M - 1)$. Pentru a face asta, i se permite să se deplaseze repetat din celula curentă (x, y) în orice altă celulă $(x + d, y)$ sau $(x, y + d)$, astfel încât $d > 0$. Pentru fiecare asemenea mutare, Simona va primi drept recompensă $|A_{x,y} - A_{x',y'}| - C$ monede, unde x', y' sunt noile sale coordonate și C este un cost fixat înainte de începutul călătoriei. Dacă rezultatul expresiei $|A_{x,y} - A_{x',y'}| - C$ este un număr negativ, Simona va pierde monede. Rețineți că este posibil ca la sfârșitul jocului numărul de monede să fie negativ.

Ajutați-o pe Simona să determine numărul maxim de monede cu care poate termina jocul.

Avem $|a| = a$, dacă $a \geq 0$ și $|a| = -a$, altfel.



Detalii de implementare

Aveți de implementat funcția `max_profit`:

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

- N, M : dimensiunile matricei;
- C : constanta fixată pentru test;
- A : vector de vectori de întregi de dimensiune $N \times M$, reprezentând matricea (indexată pe linii și pe coloane).

Această funcție trebuie executată o singură dată pentru fiecare test și trebuie să returneze profitul maxim cu care se termină jocul.



Restricții

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



Subtask-uri

Subtask	Puncte	Subtask-uri necesare	Restricții
0	0	—	Exemplul.
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	—



Example

Considerăm următorul apel:

```
find_sum(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}})
```

În acest caz drumul optim este $(0, 0) \xrightarrow{7} (0, 2) \xrightarrow{2} (1, 2) \xrightarrow{10} (1, 5) \xrightarrow{8} (4, 5)$ și profitul obținut în acest caz este $7 + 2 + 10 + 8 = 27$. Funcția trebuie să returneze 27.

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

Aici funcția trebuie să returneze: -197 . Rețineți că rezultatul poate fi negativ.



exemplu de grader

Formatul datelor de intrare este următorul:

- linia 1: trei întregi - valorile pentru N , M și C .
- liniile 2 – $(N + 1)$: M întregi - valorile elementelor $A_{i,j}$.

Formatul datelor de ieșire este următorul:

- linia 1: un întreg - valoarea returnată de execuția funcției.