



## Catfish Farm

Bu Dengklek are o fermă de somni. Ferma de somni este un iaz, fiind reprezentată de un caroiaj de dimensiuni  $N \times N$ . Fiecare celulă este un pătrat de aceeași dimensiune. Coloanele caroiajului sunt numerotate de la 0 la  $N - 1$  de la vest la est, iar liniile sunt numerotate de la 0 la  $N - 1$  de la sud la nord. Vom nota o celulă poziționată la coloana  $c$  și linia  $r$  ( $0 \leq c \leq N - 1$ ,  $0 \leq r \leq N - 1$ ) prin  $(c, r)$ .

În iaz sunt  $M$  somni, numerotați de la 0 la  $M - 1$ , distribuiți în celule **distincte**. Pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ , somnul  $i$  se află în celula  $(X[i], Y[i])$  și are greutatea  $W[i]$  grame.

Pentru a prinde somni, Bu Dengklek dorește să construiască câteva diguri. Un dig din coloana  $c$  de lungime  $k$  (pentru orice  $0 \leq c \leq N - 1$  și  $1 \leq k \leq N$ ) este un dreptunghi care se întinde de la linia 0 până la linia  $k - 1$  acoperind celulele  $(c, 0), (c, 1), \dots, (c, k - 1)$ . Pentru fiecare coloană, Bu Dengklek poate alege să construiască sau nu un dig de o lungime la alegere pe acea coloană.

Somnul  $i$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ ) poate fi prins dacă celula vecină din est sau vest este parte a unui dig și nu există un dig care să acopere celula în care se află somnul, adică dacă:

- **cel puțin una** dintre celulele  $(X[i] - 1, Y[i])$  or  $(X[i] + 1, Y[i])$  este acoperită de un dig, și
- nu există dig care să acopere celula  $(X[i], Y[i])$ .

De exemplu, să considerăm un iaz de dimensiune  $N = 5$  cu  $M = 4$  somni:

- Somnul 0 se află în celula  $(0, 2)$  și are greutatea 5 grame.
- Somnul 1 se află în celula  $(1, 1)$  și are greutatea 2 grame.
- Somnul 2 se află în celula  $(4, 4)$  și are greutatea 1 gram.
- Somnul 3 se află în celula  $(3, 3)$  și are greutatea 3 grame.

Una dintre posibilitățile în care Bu Dengklek poate construi diguri este:

Înainte ca digurile să fie construite		După ce digurile au fost construite	
4			1
3			3
2	5		
1		2	
0			
	0	1	2

  

4				1
3			3	
2	5			
1		2		
0				
	0	1	2	3

Numărul din fiecare celulă indică greutatea somnului ce se află în acea celulă. Celulele colorate sunt acoperite de diguri. În acest caz, somnii 0 (din celula (0, 2)) și 3 (din celula (3, 3)) pot fi prinși. Somnul 1 (din celula (1, 1)) nu poate fi prins, deoarece există un dig care îi acoperă locația, pe când somnul 2 (din celula (4, 4)) nu poate fi prins deoarece nu există un dig pe coloanele vecine la vest sau est.

Bu Dengklek ar vrea să construiască digurile astfel încât greutatea totală a somnurilor care pot fi prinși să fie maximă posibilă. Sarcina voastră este să aflați greutatea totală maximă a somnurilor pe care Bu Dengklek poate să îi prindă după construcția digurilor.

## Detalii de Implementare

Veți implementa următoarea procedură:

```
int64 max_weights(int N, int M, int[] X, int[] Y, int[] W)
```

- $N$ : dimensiunea iazului.
- $M$ : numărul de somni.
- $X, Y$ : tablouri unidimensionale de lungime  $M$  ce descriu locațiile somnurilor.
- $W$ : tablou unidimensional de lungime  $M$  ce descrie greutatea somnurilor.
- Procedura va returna un întreg ce reprezintă greutatea totală maximă a somnurilor pe care Bu Dengklek îi poate prinde după construcția digurilor.
- Procedura este apelată exact o singură dată.

## Exemplu

Se consideră următorul apel:

```
max_weights(5, 4, [0, 1, 4, 3], [2, 1, 4, 3], [5, 2, 1, 3])
```

Acest exemplu este ilustrat mai sus, în descrierea problemei.

După construcția digurilor conform celor descrise, Bu Dengklek poate prinde somnii 0 și 3, care au greutatea totală  $5 + 3 = 8$  grame. Deoarece nu există o altă posibilitate de a construi digurile astfel încât să fie prinși somni cu greutatea totală mai mare decât 8 grame, procedura va returna 8.

## Restricții

- $2 \leq N \leq 100\,000$
- $1 \leq M \leq 300\,000$
- $0 \leq X[i] \leq N - 1, 0 \leq Y[i] \leq N - 1$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
- $1 \leq W[i] \leq 10^9$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
- Nu există doi somni care să se afle în aceeași celulă. Altfel spus,  $X[i] \neq X[j]$  sau  $Y[i] \neq Y[j]$  (pentru fiecare  $i$  și  $j$  astfel încât  $0 \leq i < j \leq M - 1$ ).

## Subtask-uri

1. (3 puncte)  $X[i]$  este par (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
2. (6 puncte)  $X[i] \leq 1$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
3. (9 puncte)  $Y[i] = 0$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
4. (14 puncte)  $N \leq 300, Y[i] \leq 8$  (pentru fiecare  $i$  astfel încât  $0 \leq i \leq M - 1$ )
5. (21 puncte)  $N \leq 300$
6. (17 puncte)  $N \leq 3000$
7. (14 puncte) Există cel mult 2 somni în fiecare coloană.
8. (16 puncte) Fără restricții adiționale.

## Grader-ul Local

Grader-ul local citește intrarea în următorul format:

- linia 1:  $N M$
- linia  $2 + i$  ( $0 \leq i \leq M - 1$ ):  $X[i] Y[i] W[i]$

Grader-ul local afișează răspunsul vostru în următorul format:

- linia 1: valoarea returnată a `max_weights`