

Problema Park

Fișier de intrare park.in
Fișier de ieșire park.out

Harta orașului Old Non este reprezentată ca o matrice imensă cu N linii și M coloane. Liniile sunt numerotate de sus în jos, de la 0 la $N - 1$, iar coloanele de la stânga la dreapta, de la 0 la $M - 1$. O poziție pe hartă este dată printr-o pereche (x, y) , unde x este indicele liniei (poziția pe verticală), iar y este indicele coloanei (poziția pe orizontală); astfel, colțul din stânga-sus al hărții are coordonatele $(0, 0)$.

Personajul nostru, Nefe, se află în mașina sa excentrică, care ocupă o zonă dreptunghiulară pe hartă. El trebuie să parcheze mașina într-unul dintre spațiile de parcare special amenajate, evitând în același timp clădirile și alte obstacole din oraș.

Mașina lui Nefe se poate deplasa doar prin translație în sus, jos, stânga sau dreapta. Prin translație înțelegem că întreaga mașină se mută rigid, păstrându-și forma și dimensiunile (nu se rotește și nu se deformează): la o mutare, toate celulele din care este formată mașina se deplasează simultan cu o unitate în aceeași direcție.

Regulile orașului sunt stricte când vine vorba de deplasarea autovehiculelor:

- Nicio parte a mașinii nu are voie să părăsească limitele matricei: orice celulă a mașinii trebuie să aibă linia în intervalul $[0, N - 1]$ și coloana în intervalul $[0, M - 1]$.
- Nicio parte a mașinii nu se poate suprapune, nici măcar parțial, peste un obstacol.
- Pentru a fi considerată „parcată”, mașina trebuie să se afle integral în interiorul unei zone de parcare valide.

Cerință

Nefe vrea să afle numărul minim de mutări necesare pentru a duce mașina din poziția inițială într-o poziție de parcare validă. Dacă acest lucru este imposibil, se va afișa -1 .

Date de intrare

Prima linie conține 6 numere întregi separate prin spații: N , M , $start_x$, $start_y$, end_x , end_y , reprezentând numărul de linii N și numărul de coloane M ale orașului, urmate de coordonatele colțului stânga-sus, respectiv dreapta-jos, ale mașinii în poziția inițială.

A doua linie conține un număr întreg K , reprezentând numărul de obstacole.

Următoarele K linii conțin câte 4 numere întregi x_1, y_1, x_2, y_2 , reprezentând coordonatele colțului stânga-sus și dreapta-jos ale fiecărui obstacol.

Următoarea linie conține un număr întreg P , reprezentând numărul zonelor de parcare.

Următoarele P linii conțin câte 4 numere întregi x_1, y_1, x_2, y_2 , reprezentând coordonatele colțurilor stânga-sus și dreapta-jos ale fiecărei zone de parcare.

Date de ieșire

Se va afișa un singur număr întreg: numărul minim de mutări necesare pentru a parca mașina. Dacă mașina nu poate ajunge în nicio parcare validă, se va afișa -1 .

Restricții și precizări

- $1 \leq N, M \leq 10^{15}$
- $0 \leq start_x \leq end_x < N$ și $0 \leq start_y \leq end_y < M$
- $0 \leq K, P \leq 500$
- Toate coordonatele obstacolelor și parcărilor se află în interiorul limitelor orașului.
- Mașina, obstacolele și parcărilor din datele de intrare sunt toate reprezentate de dreptunghiuri **disjuncte** două câte două.
- Unele dintre valorile din datele de intrare, precum și răspunsul, pot depăși domeniul tipului întreg pe 32 de biți; se recomandă utilizarea unui tip de date pe 64 de biți (de exemplu, `long long` în C++). De asemenea, se garantează că răspunsul nu va depăși domeniul tipului de date întreg cu semn pe 64 de biți (de exemplu, `long long` în C++).

#	Punctaj	Restricții
1	17	$1 \leq N, M \leq 2\,000$, $start_x = end_x$, $start_y = end_y$
2	12	$1 \leq N, M \leq 2\,000$
3	7	$1 \leq N, M \leq 10^{15}$, $K = 0$
4	8	$1 \leq N \leq 10^{15}$, $M = 1$
5	24	$1 \leq N, M \leq 10^{15}$, $start_x = end_x$, $start_y = end_y$
6	21	$1 \leq N, M \leq 10^{15}$, $0 \leq K, P \leq 200$
7	11	Fără restricții suplimentare

Exemple

park.in	park.out
8 6 0 0 1 1 1 4 0 5 3 2 5 4 5 4 6 0 7 1	14

Explicații

Matricea are $N = 8$ linii și $M = 6$ coloane. Mașina este un dreptunghi (mai exact, un pătrat în acest caz) de 2×2 , aflat inițial cu colțul din stânga-sus în $(0, 0)$, adică în linia 0, coloana 0.

Dintre cele două parări, prima (de dimensiune 1×1) este prea mică pentru ca mașina 2×2 să încapă integral în ea, deci nu poate fi folosită. Rămâne parcare (6, 0)–(7, 1), în care mașina încapă exact.

Deplasarea directă în jos (de-a lungul coloanelor 0 și 1) este blocată de obstacolul (4, 0)–(5, 3) (liniile 4–5, coloanele 0–3). Pentru a-l ocoli, mașina se deplasează 4 coloane la dreapta (în coloanele 4–5), coboară 6 linii pentru a trece de obstacol, apoi revine 4 coloane la stânga, intrând complet în parcare. În total $4 + 6 + 4 = 14$ mutări, ceea ce reprezintă și minimumul posibil.