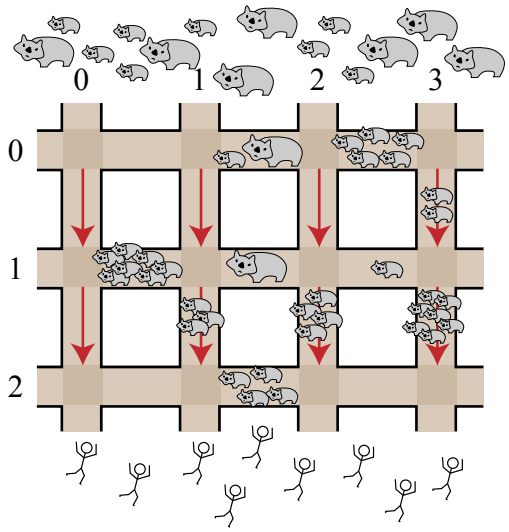
wombats

Romana — 1.1

Orașul Brisbane a fost preluat de wombați mutanți uriași și voi trebuie să conduceți oamenii zona sigură.

Străzile din Brisbane formează un grid. Există `R` străzi orizontale care merg de la est la vest, numerotate cu `0, …, (R - 1)` în ordine de la nord la sud și `C` străzi verticale care merg de la nord la sud, numerotate cu `0, …, (C - 1)` de la vest la est, ca in figura urmatoare.



Wombații au invadat din nord și oameni scapă prin sud. Oamenii pot fugi pe străzile orizontale în ambele direcții, dar pe străzile verticale ei vor putea fugi doar spre sud, spre zona sigură.

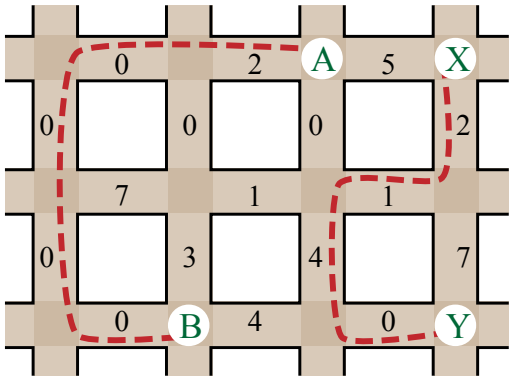
Intersecția străzii orizontale P cu strada verticală Q este notată (P,Q). Fiecare segment de stradă dintre două intersecții conține un număr de wombați și aceste numere se pot schimba în timp. Vi se cere să conduceți anumite persoane din anumite intersecții date din nord (de pe strada orizontală `0`) la anumite intersecții din sud (pe strada orizontală `R - 1`), folosind o rută care conține cât mai puțini wombați posibil.

Inițial veți primi dimensiunea gridului și numărul de wombați de pe fiecare segment de stradă. În continuare veți primi o serie de `E` evenimente din unul din următoarele două tipuri:

- *change*, care modifică numărul de wombați pe un anumit segment de strada
- *escape*, prin care o anumită persoană sosește la o amunită intersecție dată de pe strada orizontală `0` și trebuie să găsiți un traseu până la o anumită intersecție dată de pe linia `R - 1` și trece peste cât mai puțini wombați posibil.

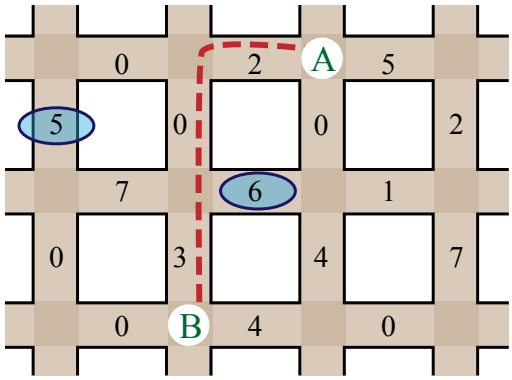
Trebuie să tratați aceste evenimente implementând rutinele `init()`, `changeH()`, `changeV()` și `escape()`, în modul descris mai jos.

Exemple



Imaginea de mai sus arată inițial o hartă cu `R = 3` drumuri orizontale si `C = 4` drumuri verticale, cu numarul de wombați marcați pe fiecare segment. Considerați următoarea serie de evenimente:

- O persoană sosește la intersecția `A = (0, 2)` și dorește să scape la intersecția `B = (2, 1)`. Numărul minim de wombați cu care se întâlnește este `2`, cum este indicat de linia punctată.
- Altă persoană sosește în intersecția `X = (0, 3)` și dorește să scape la intersecția `Y = (2, 3)`. Numărul minim de wombați cu care se întâlnește este `7`, iarăși indicat de linia punctată.
- Două evenimente de modificare apar: numărul de wombați pe segmentul de top din drumul vertical `0` se modifică în `5`, iar numărul de wombați din segmentul de mijloc al drumului orizontal `1` se modifică în `6`. Vedeți numerele încercuite din imaginea de mai jos.



- O a treia persoană sosește în `A = (0, 2)` și dorește să scape la intersecția `B = (2, 1)`. Acum numărul minim de wombați cu care se întâlnește este `5`, cum este indicat de linia punctată.

Implementare

Va trebui sa submitați un fișier implementând procedurile `init()`, `changeH()` și `changeV()` și funcția `escape()`, după cum urmează:

Procedura vostră: `init()`

C/C++
```
void init(int R, int C, int H[5000][200], int V[5000][200]);
```

Pascal
```
type wombatsArrayType = array[0..4999, 0..199] of LongInt;

procedure init(R, C : LongInt; var H, V : wombatsArrayType);
```

Descriere

Această procedură vă dă configurația inițiala a hărții, și vă permite să inițializați eventualele variabile globale și structuri de date. Va fi apelată o singură dată, înainte de orice apel către `changeH()`, `changeV()` sau `escape()`.

Parametrii

- `R`: Numărul de drumuri orizontale.
- `C`: Numărul de drumuri verticale.
- `H`: Array bidimensional de dimensiune `R × (C - 1)`, unde `H[P][Q]` vă dă numarul de wombați de pe segmentul de drum orizontal din dintre intersecțiile `(P, Q)` și `(P, Q + 1)`.
- `V`: Array bidimensional de dimensiune `(R - 1) × C`, unde `V[P][Q]` vă dă numărul de wombați de pe segmentul de drum vertical dintre intersecțiile `(P, Q)` și `(P + 1, Q)`.

Procedura voastră: `changeH()</tt:`

C/C++ `void changeH(int P, int Q, int W);`

Pascal `procedure changeH(P, Q, W: LongInt);`

Descriere

Acestă procedură va fi apelată atunci când numărul de wombați se modifică pe segmentul orizontal dintre intersecţiile (P, Q) și $(P, Q + 1)$.

Parametrii

- `<tt>P` : Indică care drum orizontal este afectat ($0 \le P \le R - 1$).
- `Q` : Indică între care două drumuri verticale se află segmentul ($0 \le Q \le C - 2$).

Procedura voastră: changeV()

C/C++
```
void changeV(int P, int Q, int W);
```

Pascal
```
procedure changeV(P, Q, W: LongInt);
```

Descriere

Acestă procedură va fi apelată când numărul de wombați se modifică pe pe segmentul de drum vertical dintre intersecțiile (P, Q) și $(P + 1, Q)$.

= Parametrii

- `P`: Indică între care două drumuri orizontale se află segmentul ($0 \le P \le R - 2$).
- `Q`: Indică care drum este afectat ($0 \le Q \le C - 1$).
- `W`: Numărul nou de wombați pe acest segment de drum ($0 \le W \le 1{,}000$).

Funcția voastră: `escape()`

C/C++ `int escape(int V1, int V2);`

Pascal `function escape(V1, V2 : LongInt) : LongInt;`

Descriere

Acestă funcție trebuie să calculeze numărul minim de wombați pe care o persoană îi va întâlni atunci când călătorește de la intersecția `(0, V1)` la `(R-1, V2)` .

Parametrii

- `V1` : Indică unde o persoană începe pe rândul orizontal 0 (`0 ≤ V1 ≤ C-1`).
- `V2` : Indică unde o persoana sfârșește pe rândul orizontal `R-1` (`0 ≤ V2 ≤ C-1`).
- *Returnează*: Numărul minim de wombați pe care persoana va fi necesar să îi întâlnească.

Exemplu de Sesiune

Următorul scenariu descrie exemplul de mai sus:

Function Call	Returns
`init(3, 4, [[0,2,5], [7,1,1], [0,4,0]], [[0,0,0,2], [0,3,4,7]])`	
`escape(2,1)`	`2`
`escape(3,3)`	`7`
`changeV(0,0,5)`	
`changeH(1,1,6)`	
`escape(2,1)`	`5`

Constrângeri

- Limită de timp: 20 secunde
- Limită de memorie: 256 MiB
- $2 \le R \le 5000$
- $1 \le C \le 200$
- Cel mult 500 de schimbări (apeluri la `changeH()` sau `changeV()`)
- Cel mult 200000 apeluri către `escape()`
- Cel mult 1000 wombați pe fiecare segment în orice moment.

Subtaskuri

Subtask	Punctaje	Constrângeri Adiționale a Inputului
1	9	$C = 1$
2	12	$R,C \le 20$, și nu vor fi apeluri către `changeH()` sau `changeV()`
3	16	$R,C \le 100$, și vor fi cel mult 100 de apeluri către `escape()`
4	18	$C = 2$
5	21	$C \le 100$
6	24	*(None)*

Testare

Graderul de pe computerul vostru va citi input din fișierul `wombats.in`, care va fi în următorul format:

- linia 1: `R C`
- linia 2: `H[0][0] … H[0][C-2]`
- …
- linia `(R + 1)`: `H[R-1][0] … H[R-1][C-2]`
- linia `(R + 2)`: `V[0][0] … V[0][C-1]`
- …
- linia `(2R)`: `V[R-2][0] … V[R-2][C-1]`
- următoarea linie: `E`
- următoarele `E` linii: un eveniment pe linie, în ordinea în care apar

Dacă `C = 1`, liniile goale conținând numărul de wombați pe drumurile orizontale (liniile `2` ... `R + 1`) nu sunt necesare.

Linia fiecărui eveniment trebuie să fie în una din formatele:

- pentru a indica `changeH(P, Q, W)`: `1 P Q W`
- pentru a indica `changeV(P, Q, W)`: `2 P Q W`
- pentru a indica `escape(V1, V2)`: `3 V1 V2`

În acest fel, exemplul de mai sus trebuie să fie dat în următorul format:

```
3 4
0 2 5
7 1 1
0 4 0
0 0 0 2
0 3 4 7
5
3 2 1
3 3 3
2 0 0 5
1 1 1 6
3 2 1
```

Note de limbaj

C/C++ Trebuie să faceți `#include "wombats.h"`.

Pascal Trebuie să definiți `unit Wombats`. Toți vectorii sunt indexați de la `0` (nu de la `1`).

Vedeți template-urile de soluții de pe calculatoarele voastre pentru exemple.