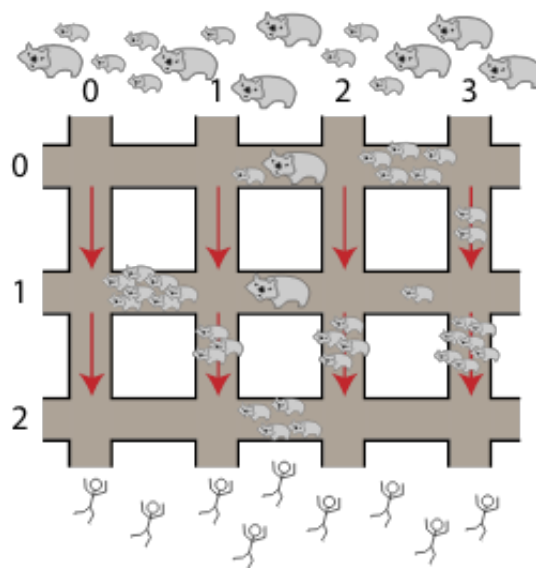


Grad Brizbejn je okupiran velikim mutiranim botovima, pa vi morate pomoći građanima da se spasu.

Putevi u Brizbejnu su oblika velike rešetke. Postoji R horizontalnih puteva koji idu od zapada ka istoku, numerisani $0, \dots, (R - 1)$ redom od severa ka jugu, i C vertikalnih puteva koji idu od severa ka jugu, numerisani $0, \dots, (C - 1)$ redom od zapada ka istoku, kao što je prikazano na slici ispod.



Botovi su napali sa severa, i ljudi pokušavaju da pobegnu na jug. Ljudi se mogu kretati horizontalnim putevima u bilo kom smeru, ali vertikalnim se mogu kretati *samo ka jugu (na dole)*.

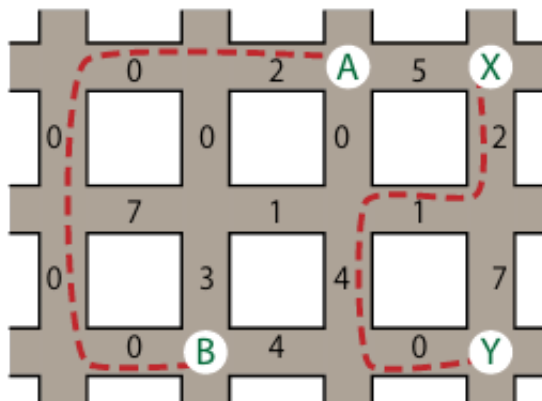
Raskrsnica horizontalnog puta P i vertikalnog puta Q je obeležena sa (P, Q) . Na svakom segmentu puta između dve raskrsnice se nalazi neki broj botova, i ti brojevi se mogu menjati vremenom. Vaš zadatak je da pomognete svakoj osobi da dođe od neke date raskrsnice koja se nalazi skroz severno (na horizontalnom putu 0) do neke date raskrsnice koja je skroz južno (na horizontalnom putu $R - 1$), tako što će ići rutom koja prolazi najmanje moguće botova.

Na početku će vam biti data veličina rešetke i broj botova na svakom segmentu puta. Nakon toga, biće vam dati redom E događaja, od kojih je svaki jedan od ove dve vrste:

- *change*, označava da se broj botova promenio na nekom segmentu; ili
- *escape*, označava da neka osoba dolazi na datu raskrsnicu na horizontalnom putu 0 , a vi morate pronaći rutu do date raskrsnice na horizontalnom putu $R - 1$ koja prolazi najmanje mogući broj botova.

Morate obraditi ove događaje implementirajući funkcije `init()`, `changeH()`, `changeV()` i `escape()`, kao što je opisano ispod.

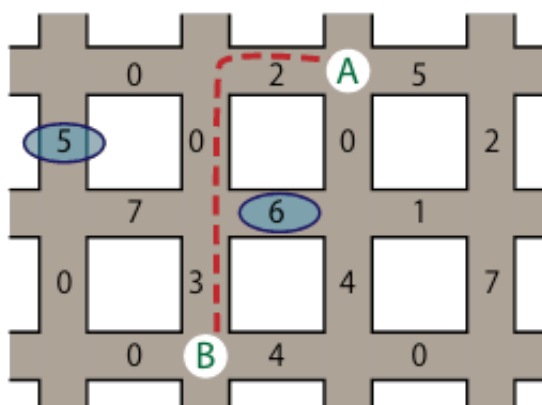
Primeri



Slika iznad pokazuje početnu mapu gde je $R = 3$ broj horizontalnih puteva i $C = 4$ broj vertikalnih puteva, a broj botova je obeležen na svakom segmentu.

Posmatrajmo sledeći redosled događaja:

- Osoba dolazi na raskrsnicu $A = (0, 2)$ i želi da pobjegne na raskrsnicu $B = (2, 1)$. Najmanji mogući broj botova koje usput može da prođe je 2 , što je obeleženo isprekidanom linijom.
- Druga osoba dolazi na raskrsnicu $X = (0, 3)$ i želi da pobjegne na raskrsnicu $Y = (2, 3)$. Najmanji mogući broj botova koje usput može da prođe je 7 , što je obeleženo isprekidanom linijom.
- Dve izmene (dogadjaj `change`) se dešavaju: broj botova na najvišem (najsevernijem) segmentu vertikalnog puta 0 se menja na 5 , i broj botova na srednjem segmentu horizontalnog puta 1 se menja na 6 . Pogledajte zaokružene brojeve na slici ispod.



- Treća osoba dolazi na raskrnicu $A = (0, 2)$ i želi da pobjegne na raskrnicu $B = (2, 1)$. Sada je najmanji mogući broj botova koji može usput da prodje 7, kao što je obeleženo isprekidanom linijom.

Implementacija

Vi treba da priložite datoteku sa implementacijom procedura `init()`, `changeH()` i `changeV()` i funkcije `escape()`, koje treba da se ponašaju kao što je opisano u nastavku:

Vaša procedura: `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);`

Opis

Ova procedura Vam daje inicijalni izgled mape i dozvoljava da inicijalizujete globalne promenljive i strukture podataka. Ova procedura se poziva samo jednom, pre bilo kog pozivanja procedura `changeH()`, `changeV()` ili funkcije `escape()`.

Argumenti (parametri)

- `R`: Broj horizontalnih puteva.
- `C`: Broj vertikalnih puteva.
- `H`: Dvodimenzioni niz (matrica) čije su dimenzije $R \times (C - 1)$, pri čemu je `H[P][Q]` broj botova na delu horizontalnog puta između raskrsnica (P, Q) i $(P, Q + 1)$.
- `V`: Dvodimenzioni niz (matrica) čije su dimenzije $(R - 1) \times C$, pri čemu je `V[P][Q]` broj botova na segmentu vertikalnog puta između raskrsnica (P, Q) i $(P + 1, Q)$.

Vaša procedura: `changeH()`

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

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

Opis

Ova procedura se poziva pri promeni broja botova na delu horizontalnog puta između raskrsnica (P, Q) i $(P, Q + 1)$.

Argumenti (parametri)

- P : redni broj horizontalnog puta na kome se menja broj botova ($0 \leq P \leq R - 1$).
- Q : govori izmedju koja dva vertikalna puta dolazi do promene ($0 \leq Q \leq C - 2$).
- W : Novi broj vobmata na tom delu puta ($0 \leq W \leq 1,000$).

Vaša procedura: `changeV()`

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

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

Opis

Ova procedura se poziva pri promeni broja botova na delu vertikalnog puta izmedju raskrsnica (P, Q) i $(P + 1, Q)$.

Argumenti (parametri)

- P : Govori izmedju kojih vertikalnih puteva dolazi do promene ($0 \leq P \leq R - 2$).
- Q : Redni broj vertikalnog puta na kome se menja broj vobmatova ($0 \leq Q \leq C - 1$).
- W : Novi broj botova na tom delu puta ($0 \leq W \leq 1,000$).

Vaša funkcija: `escape()`

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

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

Opis

Ova funkcija određuje minimalni mogući broj botova koji neka osoba mora sresti ako putuje od raskrsnice $(0, V1)$ do raskrsnice $(R-1, V2)$.

Argumenti (parametri)

- $V1$: Govori od koje raskrsnice na horizontalnom putu 0 osoba kreće ($0 \leq V1 \leq C-1$).
- $V2$: Govori o raskrsnici na horizontalnom putu $R-1$ ($0 \leq V2 \leq C-1$) na koju osoba želi da stigne.
- *Vraća*: Najmanji broj botova koje ta osoba mora sresti na putu.

Primer

Sledeći niz poziva opisuje prethodno opisani primer:

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

Ograničenja

- Vremensko ograničenje: 20 sekundi
- Memirskoograničenje: 256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Najviše 500 promena (poziva procedura `changeH()` ili `changeV()`)
- Najviše 200,000 poziva funkcije `escape()`
- Najviše 1,000 botova na bilo kom delu u bilo kom trenutku.

Podzadaci

Podzadatak	Poeni	Dodatna ograničenja vezana za ulaz
1	9	$C = 1$
2	12	$R, C \leq 20$, i nema poziva procedura <code>changeH()</code> ili <code>changeV()</code>
3	16	$R, C \leq 100$, i ima najviše 100 poziva funkcije <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Nema)

Lokalno testiranje

Grejder na Vašem računaru treba da pročita ulazne podatke iz datoteke `wombats.in`.
Datoteka ima sledeći format:

- red 1: `R C`
- red 2: `H[0][0] ... H[0][C-2]`
- ...
- red $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- red $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- red $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- sledeći red: `E`
- sledećih `E` redova: jedan događaj u svakom redu, u redosledu u kome se dešavaju

Ako je `C = 1`, prazni redovi koji sadrže broj botova na horizontalnim putevima (redovi od 2 do $R + 1$) nisu neophodni.

Svaki red za događaj ima jedan od sledećih formata:

- Da bi opisali poziv procedure `changeH(P, Q, W) : 1 P Q W`
- da bi opisali poziv procedure `changeV(P, Q, W) : 2 P Q W`
- da bi opisali poziv funkcije `escape(V1, V2) : 3 V1 V2`

Na primer, prethodno opisani primer bi bio zapisan u sledećem formatu:

```
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
```

Napomene vezane za programski jezik

C/C++ Morate dodati `#include "wombats.h"`.

Pascal Morate definisati `unit Wombats`. Svi nizovi su indeksirani od 0 (a ne 1).

Pogledajte template rešenja na vašem računaru.

