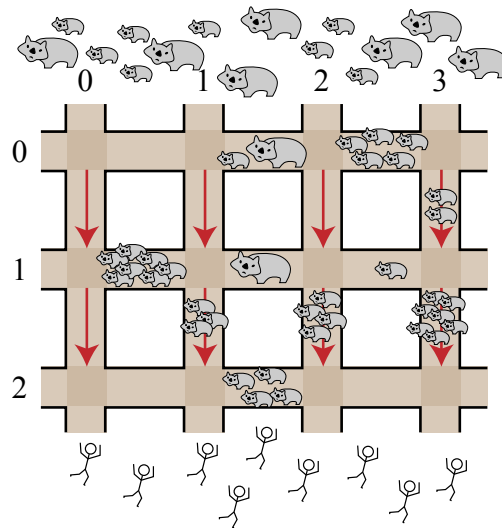


Brisbanen ovat vallanneet mutaattivompatit, ja tehtäväsi on johdattaa ihmiset turvaan.

Brisbanen katuja esittää suuri ruudukko. Siinä on R vaakasuuntaista tietä länsi-itä-suunnassa, jotka on numeroitu $0, \dots, (R - 1)$ järjestyksessä pohjoisesta etelään, sekä C pystysuuntaista tietä pohjois-etelä-suunnassa, jotka on numeroitu $0, \dots, (C - 1)$ järjestyksessä lännestä itään. Alla oleva kuva selvittää asiaa.



Vompatit saapuvat pohjoisesta, ja ihmiset pakenevat etelään. Ihmiset voivat liikkua vaakasuuntaisia teitä molempiin suuntiin, mutta he voivat liikkua pystysuuntaisia teitä *vain etelää kohti* turvaan.

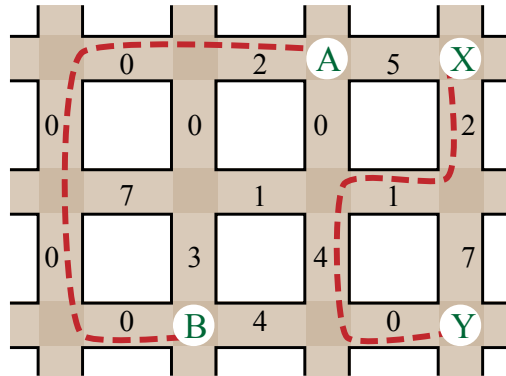
Vaakasuuntaisen tien P ja pystysuuntaisen tien Q risteys merkitään (P, Q) . Jokaisella kahden risteuksen välisellä tieosuudella on tietty määrä vompatteja, ja tämä määrä saattaa vaihdella ajan kuluessa. Tehtäväsi on johdattaa jokainen ihminen tietystä pohjoisessa olevasta risteyksestä (vaakasuuntaisella tiellä 0) tiettyyn etelässä olevaan risteykseen (vaakasuuntaisella tiellä $R - 1$) käyttäen reittiä, jonka varrella on mahdollisimman vähän vompatteja.

Aluksi sinulle annetaan ruudukon koko sekä vompattien määrä kullakin tieosuudella. Tämän jälkeen sinulle annetaan E tapahtumaa, joista jokainen on joko:

- *muutos*, joka vaihtaa vompattien määrän jollain tieosuudella, tai
- *pako*, jossa ihminen saapuu tiettyyn risteykseen vaakasuuntaisella tiellä 0 , ja sinun täytyy etsiä sellainen reitti tiettyyn risteykseen vaakasuuntaisella tiellä $R - 1$, jonka varrella on mahdollisimman vähän vompatteja.

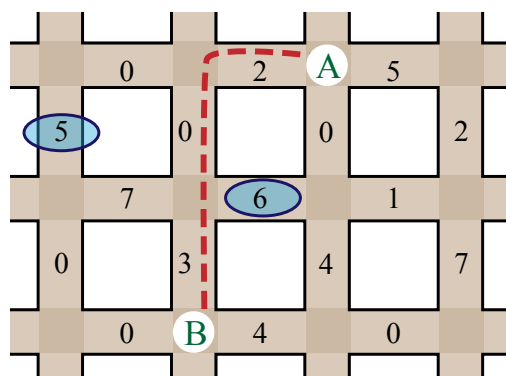
Sinun täytyy käsitellä nämä tapahtumat toteuttamalla funktiot `init()`, `changeH()`, `changeV()` ja `escape()` alla olevan mukaisesti.

Esimerkit



Yllä olevassa kuvassa on alkutilanteessa oleva kartta, jossa on $R = 3$ vaakasuuntaista tietä sekä $C = 4$ pystysuuntaista tietä sekä vompattien määrä kullakin tieosuudella. Tarkastellaan seuraavaa tapahtumasarjaa:

- Ihminen saapuu risteykseen $A = (0, 2)$ ja haluaa paeta risteykseen $B = (2, 1)$. Pienin mahdollinen vompattien määrä reitin varrella on 2 katkoviivan mukaisesti.
- Toinen ihminen saapuu risteykseen $X = (0, 3)$ ja haluaa paeta risteykseen $Y = (2, 3)$. Pienin mahdollinen vompattien määrä reitin varrella on 7 jälleen katkoviivan mukaisesti.
- Kaksi muutosta tapahtuu: vompattien määrä pystysuuntaisen tien 0 ylimmällä osuudella on nyt 5 , ja vompattien määrä vaakasuuntaisen tien 1 keskiosuudella on nyt 6 . Nämä kohdat on ympyröity alla olevassa kuvassa.



- Kolmas ihminen saapuu risteykseen $A = (0, 2)$ ja haluaa paeta risteykseen $B = (2, 1)$. Nyt pienin vompattien määrä on 5 uuden katkoviivan mukaisesti.
-

Toteutus

Sinun täytyy lähettää tiedosto, jossa on toteutettuna funktiot `init()`, `changeH()`, `changeV()` ja `escape()` seuraavasti:

Sinun funktiosi: `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);`

Kuvaus

Tämän funktion kautta saat kartan aloitustilanteen, ja voit alustaa mitä tahansa globaaleita muuttujia ja tietorakenteita. Funktiota kutsutaan vain kerran ennen kaikkia kutsuja funktioihin `changeH()`, `changeV()` ja `escape()`.

Parametrit

- `R`: Vaakasuuntaisten teiden määrä.
- `C`: Pystysuuntaisten teiden määrä.
- `H`: Kaksiulotteinen taulukko kokoa $R \times (C - 1)$, jossa `H[P][Q]` on vampattien määrä vaakasuuntaisella tieosuudella risteysten (P, Q) ja $(P, Q + 1)$ välillä.
- `V`: Kaksiulotteinen taulukko kokoa $(R - 1) \times C$, jossa `V[P][Q]` on vampattien määrä pystysuuntaisella tieosuudella risteysten (P, Q) ja $(P + 1, Q)$ välillä.

Funktiosi: changeH()

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

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

Kuvaus

Tätä funktiota kutsutaan, kun vompattien määrä muuttuu vaakasuuntaisella tieosuudella risteysten (P, Q) ja $(P, Q + 1)$ välillä.

Parametrit

- P : Vaakasuuntainen tie, jossa muutos tapahtuu ($0 \leq P \leq R - 1$).
- Q : Kaksi pystysuuntaista tietä ($0 \leq Q \leq C - 2$), joiden välissä tieosuus sijaitsee.
- W : Uusi vompattien määrä tällä tieosuudella ($0 \leq W \leq 1\,000$).

Sinun funktiosi: `changeV()`

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

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

Kuvaus

Tätä funktiota kutsutaan, kun vompattien määrä muuttuu pystysuuntaisella tieosuudella risteysten (P, Q) ja $(P + 1, Q)$ välillä.

Parametrit

- P : Kaksi vaakasuuntaista tietä ($0 \leq P \leq R - 2$), joiden välissä tieosuus sijaitsee.
- Q : Pystysuuntainen tie, jossa muutos tapahtuu ($0 \leq Q \leq C - 1$).
- W : Uusi vompattien määrä tällä tieosuudella ($0 \leq W \leq 1\,000$).

Sinun funktiosi: `escape()`

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

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

Kuvaus

Tämän funktio tulee laskea pienin mahdollinen vompattien määrä reitillä risteyksestä $(0, V1)$ risteykseen $(R-1, V2)$.

Parametrit

- $V1$: Ihmisen aloituskohta vaakarivillä 0 ($0 \leq V1 \leq C-1$).
- $V2$: Ihmisen lopetuskohta vaakarivillä $R-1$ ($0 \leq V2 \leq C-1$).
- *Palauttaa*: Pienin vompattien määrä reitin varrella.

Esimerkki-istunto

Seuraava istunto vastaa edellä ollutta esimerkkiä:

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

Rajat

- Aikaraja: 15 sekuntia
- Muistiraja: 256 MiB
- $2 \leq R \leq 5000$
- $1 \leq C \leq 200$
- Korkeintaan 500 funktioiden `changeH()` ja `changeV()` kutsua
- Korkeintaan 200000 funktion `escape()` kutsua
- Korkeintaan 1000 vompattia tietyllä tieosuudella tietyllä hetkellä

Alitehtävät

Alitehtävä	Pisteet	Syötteen lisärajoitukset
1	9	$C = 1$
2	12	$R, C \leq 20$ ja ei kutsuja funktioihin <code>changeH()</code> tai <code>changeV()</code>
3	16	$R, C \leq 100$ ja korkeintaan 100 kutsua funktioon <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Ei mitään)

Kokeilu

Esimerkkiarvostelija lukee syötteen tiedostosta `wombats.in`, jonka muodon tulee olla seuraava:

- rivi 1: `R C`
- rivi 2: `H[0][0] ... H[0][C-2]`
- ...
- rivi $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- rivi $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- rivi $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- seuraava rivi: `E`
- seuraavat `E` riviä: yksi tapahtuma joka rivillä niiden tapahtumisjärjestyksessä

Jos $C = 1$, ei ole tarpeen merkitä tyhjiä rivejä, jotka kuvaisivat vompattien määrä vaakasuuntaisilla teillä (rivistä 2 riviin $R + 1$).

Jokainen tapahtuma tulee ilmoittaa seuraavanmuotoisella rivillä:

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

Esimerkiksi yllä oleva esimerkki tulisi antaa seuraavassa muodossa:

```
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
```

Huomioita kielistä

C/C++ You must `#include "wombats.h"`.

Pascal You must define the `unit Wombats`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.