



International Olympiad in Informatics 2013

6-13 July 2013

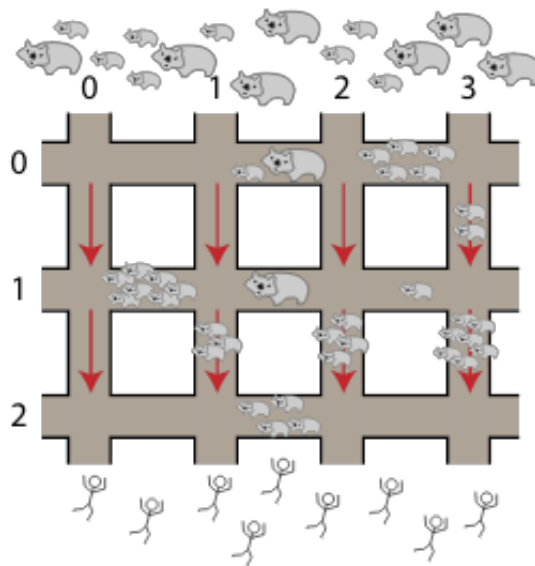
Brisbane, Australia

wombats

Lithuanian – 1.1

Milžinai mutantai vombatai plūsta į Brisbaną, o jūs turite gelbėti jo gyventojus.

Brisbano keliai sudaro didelį tinklą. R horizontalių kelių eina iš rytų į vakarus ir yra numeruojami iš šiaurės į pietus nuo 0 iki $(R - 1)$. C vertikalių kelių, eina iš šiaurės į pietus ir yra numeruojami iš vakarų į rytus nuo 0 iki $(C - 1)$. Žiūrėkite į žemiau esančią iliustraciją.



Vombatai plūsta iš šiaurės, todėl miestiečiai bėga į pietus. Horizontaliais keliais žmonės gali bėgti abejomis kryptis, tačiau vertikaliais — tik į pietus.

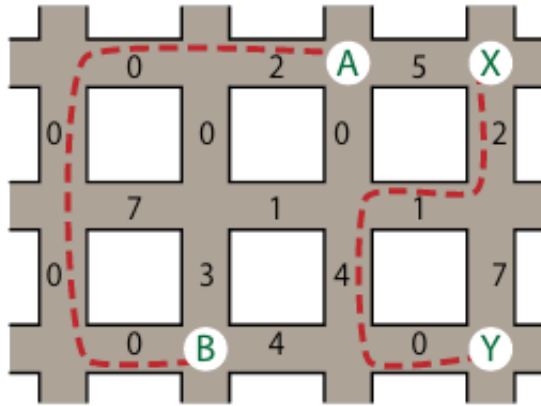
Sankryža, kurioje kertasi P -asis horizontalus ir Q -asis vertikalus kelias, žymima (P, Q) . Kiekvienoje dvi sankryšos jungiančioje kelio atkarpoje gali būti vombatų, ir bėgant laikui jų kiekis gali keistis. Jūsų užduotis — nuvesti kiekvieną žmogų iš nurodytos sankryžos šiaurėje (0 -ajame horizontalaus kelio) iki nurodytos sankryžos pietuose ($(R-1)$ -ajame horizontaliame kelyje) taip, kad tektų sutikti kuo mažiau vombatų.

Jums bus pateiktas tinklėlio dydis ir pradiniu momentu kiekvienoje kelio atkarpoje esančių vombatų skaičius. Po to bus pateikiama E dviejų rūšių nurodymų:

- `pokytis` — nurodytoje kelio atkarpoje pakinta vombatų kiekis;
- `pabėgimas` — žmogus yra nurodytoje 0 -ojo horizontalaus kelio sankryžoje, o jūs turite nuvesti jį iki nurodytos $(R-1)$ -ojo horizontalaus kelio sankryžos. Sutinkamų vombatų kiekis turi būti mažiausias įmanomas.

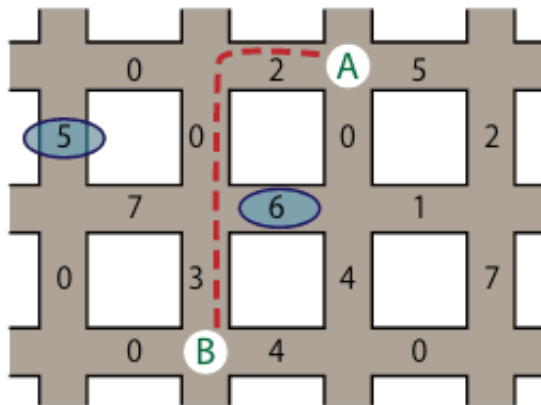
Vykdykite nurodymus realizuodami žemiau aprašytas procedūras `init()`, `changeH()`, `changeV()` ir `escape()`.

Pavyzdžiai



Piešinyje vaizduojamoje pradinėje situacijoje yra $R = 3$ horizontalių ir $C = 4$ vertikalių kelių. Kiekvienoje kelio atkarpoje įrašytas joje tuo metu esančių vombatų skaičius. Galima tokia nurodymų seka.

- Žmogus atsiranda sankryžoje $A = (0, 2)$ ir turi atsidurti sankryžoje $B = (2, 1)$. Jis gali sutikti mažiausiai 2 vombatus, kaip pavaizduota punktyrine linija.
- Kitas žmogus atsiranda sankryžoje $X = (0, 3)$ ir turi atsidurti sankryžoje $Y = (2, 3)$. Jis gali sutikti mažiausiai 7 vombatus, kaip pavaizduota kita punktyrine linija.
- Gaunami du nurodymai paeilui atlikti pakeitimus. Juos atlikus viršutinėje 0-ojo vertikalaus kelio atkarpoje yra 5 vombatai, o vidurinėje 1-ojo horizontalaus kelio atkarpoje — 6 vombatai. Tai vaizduoja apvesti skaičiai žemiau esančiame piešinyje.



- Trečias žmogus atsiranda sankryžoje $A = (0, 2)$ ir turi patekti į sankryžą $B = (2, 1)$. Jis gali sutikti mažiausiai 5 vombatus, kaip vaizduojama punktyrine linija.

Realizavimas

Pateikite failą, realizuojantį žemiau apibūdintas procedūras `init()`, `changeH()` ir `changeV()` ir funkciją `escape()`.

Jūsų procedūra: `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);`

Veikimas

Ši procedūra perduoda pradinį žemėlapi ir leidžia inicializuoti globalius kintamuosius ir duomenų struktūras. Ji kviečiama vieną kartą prieš kviečiant `changeH()`, `changeV()` ir `escape()`.

Parametrai

- `R` : horizontalių kelių skaičius.
- `C` : vertikalinių kelių skaičius.
- `H` : $R \times (C - 1)$ dydžio dvimatis masyvas; `H[P][Q]` yra vombatų skaičius horizontalioje atkarpoje tarp sankryžų `(P, Q)` ir `(P, Q + 1)`.
- `V` : $(R - 1) \times C$ dydžio dvimatis masyvas; `V[P][Q]` yra vombatų skaičius vertikalioje atkarpoje tarp sankryžų `(P, Q)` and `(P + 1, Q)`.

Jūsų procedūra: `changeH()`

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

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

Veikimas

Ši procedūra bus kviečiama, kai pakis vombatų skaičius horizontalioje atkarpoje tarp sankryžų `(P, Q)` ir `(P, Q + 1)`.

Parametrai

- `P` : nurodo kuriame horizontaliame kelyje yra atkarpa ($0 \leq P \leq R - 1$).
- `Q` : nurodo, tarp kurių vertikalinių kelių yra atkarpa ($0 \leq Q \leq C - 2$).
- `W` : pakitęs vombatų skaičius atkarpoje ($0 \leq W \leq 1,000$).

Jūsų procedūra: `changeV()`

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

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

Veikimas

Ši procedūra bus kviečiama, kai pakis vombatų skaičius vertikaliajoje atkarpoje tarp sankryžų (P, Q) ir $(P + 1, Q)$.

Parametrai

- P : nurodo, tarp kurių horizontalių kelių yra atkarpa ($0 \leq P \leq R - 2$).
- Q : nurodo, kuriame vertikaliame kelyje yra atkarpa ($0 \leq Q \leq C - 1$).
- W : pakitęs vombatų skaičius atkarpoje ($0 \leq W \leq 1,000$).

Jūsų funkcija: `escape()`

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

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

Veikimas

Ši funkcija turi apskaičiuoti mažiausią vombatų skaičių, kurį sutiks žmogus keliaujantis iš $(0, V1)$ į $(R-1, V2)$.

Parametrai

- $V1$: nurodo, kurioje vietoje 0-ame kelyje pradiniu momentu atsidurs ($0 \leq V1 \leq C-1$).
- $V2$: nurodo, kurioje vietoje $(R-1)$ -ame kelyje turi atsidurti žmogus ($0 \leq V2 \leq C-1$).
- *return*: mažiausias vombatų, kuriuos žmogui teks praeiti, skaičius.

Pavyzdžiai

Žemiau pateikiama sesija, kurioje naudojamas aukščiau pateiktas pavyzdys:

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

Ribojimai

- Laiko ribojimas: 20 sekundžių
- Atminties ribojimas: 256 MiB (MiB tas pats kas MB)
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Ne daugiau 500 pakeitimų (kreipinių į `changeH()` arba `changeV()`)
- Ne daugiau 200,000 kreipinių į `escape()`
- Ne daugiau 1,000 vombatų bet kurioje atkarpoje bet kuriuo laiko momentu

Dalinės užduotys

Dalinė užduotis	Taškai	Papildomi duomenų ribojimai
1	9	$C = 1$
2	12	$R, C \leq 20$, nebus kreipinių į <code>changeH()</code> arba <code>changeV()</code>
3	16	$R, C \leq 100$, bus ne daugiau kaip 100 kreipinių į <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Nėra)

Ekspertimentavimas

Jūsų kompiuteryje įrašytas pavyzdinis vertintojas duomenis skaitys iš failo `wombats.in` tokiu formatu:

- 1-a eilutė: `R C`
- 2-a eilutė: `H[0][0] ... H[0][C-2]`
- ...
- $(R + 1)$ -a eilutė: `H[R-1][0] ... H[R-1][C-2]`
- $(R + 2)$ -a eilutė: `V[0][0] ... V[0][C-1]`
- ...
- $(2R)$ -a eilutė: `V[R-2][0] ... V[R-2][C-1]`
- tolesnė eilutė: `E`
- tolesnės `E` eilučių: po vieną nurodymą eilutėje tokia tvarka kokia gaunami nurodymai

Jei `C = 1`, tuomet tuščios eilutės, kuriose turėtų būti vombatų kiekis horizontaliuose keliuose (jų numeriai nuo `2` iki `R + 1`), nereikalingos.

Kiekvienam nurodymui skirta eilutė turi būti aprašyta vienu šių formatų:

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

Aukščiau pateiktas pavyzdys turi būti užrašytas tokiu 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
```

Pastabos apie programavimo kalbas

C/C++ Įtraukite `#include "wombats.h"`.

Pascal Apibrėžkite `unit Wombats`. Visi masyvai numeruojami nuo `0` (ne nuo `1`).

Naudokite pateiktą sprendimo šabloną kaip pavyzdį.