



International Olympiad in Informatics 2013

6-13 July 2013

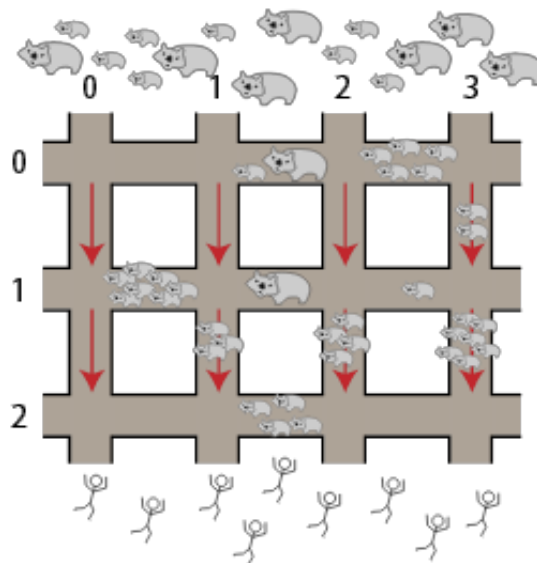
Brisbane, Australia

wombats

English — 1.1

ქალაქი ბრისბენი დაიპყრეს უზარმაზარმა მუტანტებმა, რომლებსაც ვომბატებს ეძახიან. თქვენ უნდა მოახდინოთ ადამიანების უსაფრთხო ევაკუაცია.

ბრისბენის გზები წარმოადგენენ ბადეს. ბადე შედგება R ცალი ორმხრივი ჰორიზონტალური გზისაგან, რომლებიც გადანომრილია რიცხვებით $0, \dots, (R - 1)$ ჩრდილოეთიდან სამხრეთისაკენ და C ცალი ვერტიკალური გზისაგან, რომლებიც გადანომრილია რიცხვებით $0, \dots, (C - 1)$ დასავლეთიდან აღმოსავლეთისაკენ, როგორც ეს ნაჩვენებია ნახაზზე ქვემოთ:



ვომბატები უტევენ ჩრდილოეთიდან, ხოლო ხალხი გადაადგილდება სამხრეთისაკენ. ადამიანებს შეუძლიათ იმოძრაონ ჰორიზონტალურ გზებზე.

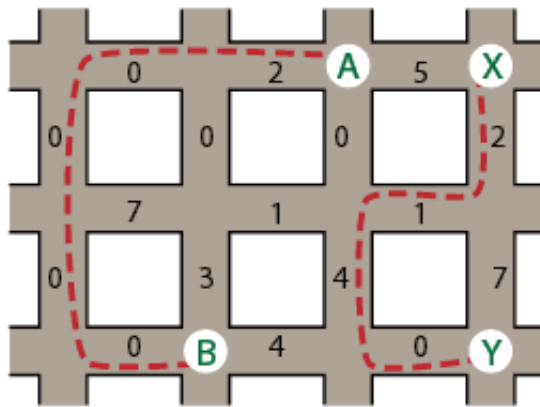
ჰორიზონტალური P გზის გადაკვეთა ვერტიკალურ Q გზასთან აღვნიშნოთ წერტილით (P, Q) . ორ წერტილს შორის გზის სეგმენტი მოიცავს ვომბატების გარკვეულ რაოდენობას, რომლებიც შეიძლება დროში შეიცვალოს. თქვენი ამოცანაა მართოთ ადამიანების ევაკუაცია ყველაზე ჩრდილოეთით (ანუ 0 ნომრის ჰორიზონტალიდან) განლაგებული გზის რომელიმე მოცემული წერტილიდან ყველაზე სამხრეთით (ანუ $R - 1$ ნომრის მქონე ჰორიზონტალამდე) განლაგებული გზის რომელიმე მოცემულ წერტილამდე ისე, რომ მოძრაობისას შევხვდეთ ვომბატების მინიმალურ ჯამურ რაოდენობას.

თავდაპირველად თქვენ მოგეცემათ ბადის ზომები და გზების თითოეულ სეგმენტზე ვომბატების საწყისი რაოდენობა. შემდეგ მოგეცემათ E რაოდენობის მოვლენა ორიდან ერთ-ერთი სახით:

- *change*, რომელიც ცვლის ვომბატების რაოდენობას გზის რომელიღაც სეგმენტზე ;
- *escape*, სადაც რომელიმე ადამიანი გადაადგილდება 0 ნომრის კორიზონტალის მოცემული წერტილიდან $R - 1$ კორიზონტალის მოცემულ წერტილამდე თქვენს მიერ შერჩეული მარშრუტით ისე, რომ გზად შეხვდეს ვომბატების მინიმალური ჯამური რაოდენობა.

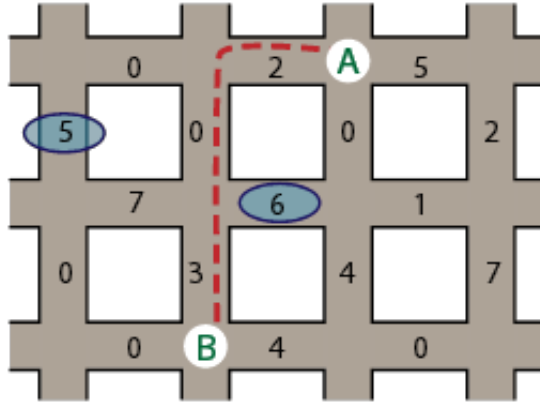
თქვენ უნდა მართოთ ეს მოვლენები შემდეგი ქვეპროგრამების საშუალებით: `init()`, `changeH()`, `changeV()` და `escape()`, როგორც ეს ქვემოთაა აღწერილი.

მაგალითი



ზემოთ მოცემულ ნახატზე აღწერილია საწყისი რუკა $R = 3$ კორიზონტალური გზითა და $C = 4$ ვერტიკალური გზით. ვომბატების რაოდენობა მოცემულია გზის თითოეული სეგმენტისთვის. განვიხილოთ მოვლენების შემდეგი სერია:

- ადამიანი გადაადგილდება $A = (0, 2)$ წერტილიდან და სურს მივიდეს $B = (2, 1)$ წერტილში. გზად შეხვედრილი ვომბატების მინიმალური ჯამური რაოდენობაა 2, რაც ნახატზე ნაჩვენებია წყვეტილი ხაზით.
- სხვა ადამიანს სურს მივიდეს $X = (0, 3)$ წერტილიდან $Y = (2, 3)$ წერტილამდე. ვომბატების მინიმალური ჯამური რაოდენობა, რომელიც ამ ორ წერტილს შორის გადაადგილებისას შეიძლება მას შეხვდეს, არის 7. შესაბამისი მარშრუტი ნახატზე ასევე ნაჩვენებია წყვეტილი ხაზით.
- ხდება ორი *change* მოვლენა: ნულოვანი ვერტიკალური გზის ზედა სეგმენტში ვომბატების რაოდენობა გახდა 5, და ნომერი 1 კორიზონტალური გზის შუა სეგმენტში ვომბატების რაოდენობა გახდა 6.



- მესამე ადამიანი მიდის $A = (0, 2)$ წერტილიდან $B = (2, 1)$ წერტილამდე. მომხდარი ცვლილებების შემდეგ გზად შეხვედრილი ვომბატების მინიმალური ჯამური რაოდენობაა 5. ნახატზე მარშრუტი ნაჩვენებია წყვეტილი ხაზით.

იმპლემენტაცია

თქვენ უნდა ატვირთოთ ფაილი შემდეგი ქვეპროგრამების იმპლემენტაციით: `init()`, `changeH()`, `changeV()` და `escape()`, როგორც ეს ქვემოთაა აღწერილი.

თქვენი პროცედურა: `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);
```

აღწერა

ეს პროცედურა გაძლევთ საწყის რუკას და ახდენს გლობალური ცვლადების და მონაცემთა სტრუქტურების ინიციალიზაციას. ის უნდა გამოიძახოს მხოლოდ ერთხელ, ვიდრე გამოიძახებთ `changeH()`, `changeV()` ან `escape()` პროცედურას.

პარამეტრები

- `R`: ჰორიზონტალური გზების რაოდენობა.
- `C`: ვერტიკალური გზების რაოდენობა..

- **H**: ორგანზომილებიანი მასივი ზომით $R \times (C - 1)$, სადაც $H[P][Q]$ გვამლევს ვომბატების რაოდენობას ჰორიზონტალური გზის სეგმენტზე (P, Q) და $(P, Q + 1)$ წერტილებს შორის.
- **V**: ორგანზომილებიანი მასივი ზომით $(R - 1) \times C$, სადაც $V[P][Q]$ გვამლევს ვომბატების რაოდენობას ვერტიკალური გზის სეგმენტზე (P, Q) და $(P + 1, Q)$ წერტილებს შორის.

თქვენი პროცედურა: **changeH()**

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

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

აღწერა

ეს პროცედურა გამოიძახება, როცა იცვლება ვომბატების რაოდენობა ჰორიზონტალური გზის სეგმენტზე (P, Q) და $(P, Q + 1)$ წერტილებს შორის.

პარამეტრები

- **P**: განსაზღვრავს ჰორიზონტალური გზის ნომერს, რომელზეც მოხდა ცვლილება ($0 \leq P \leq R - 1$).
- **Q**: განსაზღვრავს ორი ვერტიკალური გზის ნომერს, რომელთა შორის მოთავსებულ სეგმენტზე მოხდა ცვლილება ($0 \leq Q \leq C - 2$).
- **W**: მითითებულ სეგმენტზე ვომბატების ახალი რაოდენობა ($0 \leq W \leq 1,000$).

თქვენი პროცედურა: **changeV()**

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

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

აღწერა

ეს პროცედურა გამოიძახება, როცა იცვლება ვომბატების რაოდენობა ვერტიკალური გზის სეგმენტზე (P, Q) და $(P + 1, Q)$ წერტილებს შორის.

პარამეტრები

- **P**: განსაზღვრავს ორი ჰორიზონტალური გზის ნომერს, რომელთა შორის მოთავსებულ სეგმენტზე მოხდა ცვლილება ($0 \leq P \leq R - 2$).

- Q : განსაზღვრავს ვერტიკალური გზის ნომერს, რომელზეც მოხდა ცვლილება ($0 \leq Q \leq C - 1$).
- W : მითითებულ სეგმენტზე ვომბატების ახალი რაოდენობა ($0 \leq W \leq 1,000$).

თქვენი ფუნქცია: **escape ()**

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

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

აღწერა

ამ ფუნქციამ უნდა გამოითვლოს ვომბატების უმცირესი ჯამური რაოდენობა, რომლებიც ადამიანს შეხვდება გზად $(0, V1)$ წერტილიდან $(R-1, V2)$ წერტილამდე მოძრაობისას.

პარამეტრები

- $V1$: განსაზღვრავს ნულოვანი ჰორიზონტალური გზის წერტილს ($0 \leq V1 \leq C-1$), საიდანაც ადამიანი იწყებს მოძრაობას.
- $V2$: განსაზღვრავს ჰორიზონტალური $R-1$ გზის რომელ ($0 \leq V2 \leq C-1$) წერტილში უნდა დაამთავროს მოძრაობა ადამიანმა.
- *Returns*: უნდა დააბრუნოს გზად შეხვედრილი ვომბატების მინიმალური ჯამური რაოდენობა.

შესატანი მონაცემების ნიმუში

ქვემოთ მოყვანილი ნიმუში აღწერს ზემოთ განხილულ მაგალითს:

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

შეზღუდვები

- დროის ლიმიტი: 20 წამი
- მეხსიერების ლიმიტი: 256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- არაუმეტეს 500 ცვლილებისა (გამოიძახებს ერთ–ერთს: `changeH()` ან `changeV()`)
- არაუმეტეს 200,000 გამოიძახებისა `escape()`
- არაუმეტეს 1,000 ვომბატი თითოეულ სეგმენტზე ნებისმიერ დროს.

ქვეამოცანები

ქვეამოცანა	ქულა	დამატებითი შეზღუდვა
1	9	$C = 1$
2	12	$R, C \leq 20$, და არ გამოიძახება <code>changeH()</code> ან <code>changeV()</code>
3	16	$R, C \leq 100$, და <code>escape()</code> –ის არაუმეტეს 100 გამოიძახებისა
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(None)

ექსპერიმენტირება

სანიმუშო გრადერი თქვენს კომპიუტერზე შესატან მონაცემებს წაიკითხავს `wombats.in` ფაილიდან, რომელსაც შემდეგი ფორმატი უნდა ჰქონდეს:

- სტრიქონი 1: `R C`
- სტრიქონი 2: `H[0][0] ... H[0][C-2]`
- ...
- სტრიქონი $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- სტრიქონი $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- სტრიქონი $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- შემდეგი სტრიქონი: `E`

- შემდეგი E ცალი სტრიქონი: თითოეულ სტრიქონში მოცემულია თითო მოვლენა მათი მოხდენის რიგითობის მიხედვით.

თუ $C = 1$, ჰორიზონტალური სეგმენტები არ არსებობს.

სტრიქონი თითოეული მოვლენისათვის შემდეგი ფორმატებიდან ერთ–ერთი სახით არის მოცემული:

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

მაგალითად, ზემოთ მოყვანილი მაგალითი წარმოდგენილი უნდა იყოს შემდეგი ფორმატით:

```

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

```

შენიშვნები პროგრამულ ენებზე

C/C++ თქვენ უნდა გამოიყენოთ `#include "wombats.h"`.

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

მაგალითებისათვის ამოხსნათა შაბლონები იხილეთ თქვენს კომპიუტერზე.