



## International Olympiad in Informatics 2013

6-13 July 2013

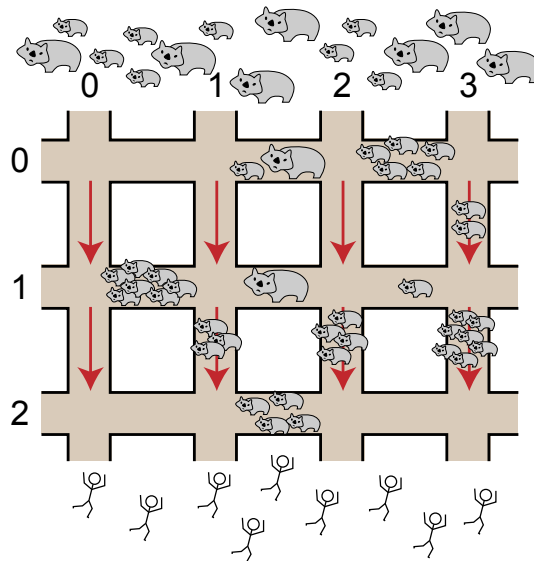
Brisbane, Australia

# wombats

Greek — 1.0

Η πόλη του Brisbane έχει καταληφθεί από μεταλλαγμένα τερατα και αποστολή σας είναι να οδηγήσετε τους ανθρώπους σε ασφαλές μέρος.

Οι δρόμοι του Brisbane έχουν τη διάταξη ενός πλέγματος. Υπάρχουν  $R$  οριζόντιοι δρόμοι που πηγαίνουν από ανατολικά προς δυτικά αριθμημένοι  $0, \dots, (R - 1)$  με αρίθμηση από βορρά προς νότο, και  $C$  και  $C$  κάθετοι δρόμοι που πηγαίνουν από βόρεια προς νότια, αριθμημένοι  $0, \dots, (C - 1)$  με αρίθμηση από δυτικά προς ανατολικά, όπως φαίνεται και στο πιο κάτω σχήμα.



Τα τερατα έχουν εισβάλει από τα βόρεια, και οι άνθρωποι διαφεύγουν προς τα νότια. Οι άνθρωποι μπορούν να τρέξουν κατα μήκος των οριζόντιων δρόμων προς οποιαδήποτε κατεύθυνση, αλλά στους κάθετους μπορούν να τρέχουν *μόνο προς τα νότια*, προς την ασφαλή περιοχή.

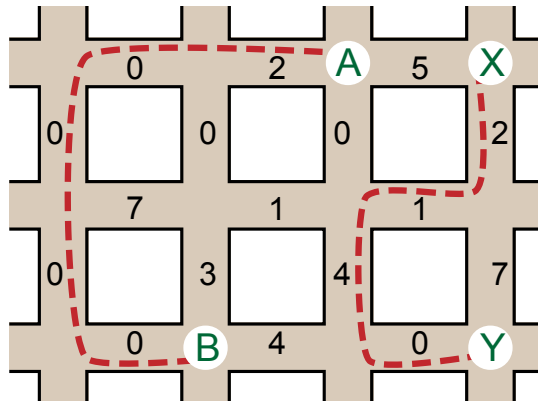
Η διασταύρωση ενός οριζόντιου δρόμου  $P$  με ένα κατακόρυφο  $Q$  δηλώνεται με  $(P, Q)$ . Κάθε τμήμα δρόμου ανάμεσα σε δύο διασταυρώσεις περιέχει κάποιο αριθμό από τερατα ο οποίος μπορεί να αυξομειώνεται με την πάροδο του χρόνου. Αποστολή σας είναι να οδηγήσετε τον κάθε ένα άνθρωπο από κάποια δεδομένη διασταύρωση στα βόρεια (στον οριζόντιο δρόμο  $0$ ) σε μια συγκεκριμένη διασταύρωση στα νότια (στον οριζόντιο δρόμο  $R - 1$ ), ακολουθώντας τη διαδρομή μέσα από την οποία θα συναντήσει όσο το δυνατό λιγότερα τερατα.

Αρχικά θα σας δοθεί το μέγεθος του πλέγματος και ο αριθμός των τεράτων σε κάθε τμήμα δρόμου. Ακολουθώς θα σας δοθεί μια σειρά από  $E$  περιστατικά το καθένα από τα οποία είναι ένα από τα πιο κάτω:

- *change*, μια αλλαγή, όπου αλλάζει ο αριθμός των τεράτων σε ένα τμήμα δρόμου, ή
- *escape*, μια διαφυγή, όπου ένας άνθρωπος φτάνει σε μια συγκεκριμένη διασταύρωση στον οριζόντιο δρόμο 0, και έσεις θα πρέπει να βρείτε τη διαδρομή προς μια συγκεκριμένη διασταύρωση στον οριζόντιο δρόμο R-1 μέσα από την οποία συναντά τον μικρότερο αριθμό από τέρατα.

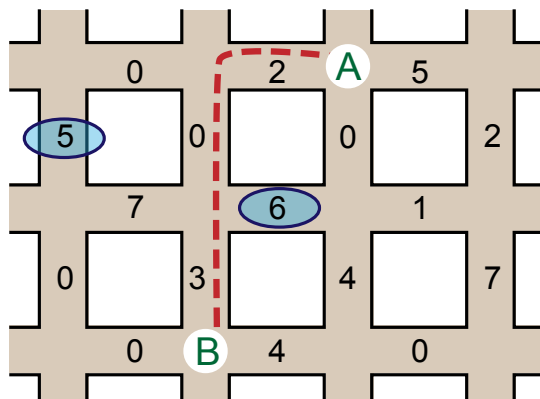
Θα πρέπει να υλοποιήσετε τα πιο πάνω γράφοντας τα ακόλουθα υποπρογράμματα `init()`, `changeH()`, `changeV()` and `escape()`, όπως ακριβώς περιγράφονται πιο κάτω.

## Παραδείγματα



Το πιο πάνω διάγραμμα δείχνει ένα αρχικό χάρτη με 3 οριζόντιους δρόμους  $R = 3$  και 4 κάθετους δρόμους  $C = 4$  ενώ ο αριθμός των τεράτων σε κάθε τμήμα δρόμου είναι αντίστοιχα σημειωμένος. Θεωρήστε τα πιο κάτω γεγονότα:

- Ένα άτομο καταφτάνει στη διασταύρωση  $A = (0, 2)$  και επιθυμεί να διαφύγει προς τη διασταύρωση  $B = (2, 1)$ . Ο μικρότερος αριθμός από τέρατα τον οποίο μπορεί να συναντήσει είναι 2, όπως φαίνεται και από τη διακεκομμένη γραμμή.
- Άλλο ένα άτομο φτάνει στη διασταύρωση  $X = (0, 3)$  και επιθυμεί να διαφύγει μέσω της διασταύρωσης  $Y = (2, 3)$ . Ο μικρότερος αριθμός τεράτων τα οποία μπορεί να συναντήσει είναι 7, όπως και πάλι φαίνεται και από τη διακεκομμένη γραμμή.
- Ακολούθως συμβαίνουν τα εξής 2 περιστατικά: ο αριθμός των τεράτων στο πιο ψηλό τμήμα του κάθετου δρόμου 0 αλλάζει σε 5, και ο αριθμός των τεράτων στο μεσαίο τμήμα του οριζόντιου δρόμου 1 αλλάζει σε 6. Οι αλλαγές σημειώνονται μέσα σε κύκλο στο πιο κάτω σχήμα.



- Ένα τρίτο άτομο φτάνει στη διασταύρωση  $A = (0, 2)$  και επιθυμεί να διαφύγει μέσω της διασταύρωσης  $B = (2, 1)$ . Τώρα ο μικρότερος αριθμός των τεράτων που θα συναντήσει είναι 5, όπως φαίνεται και από τη διακεκομμένη διαδρομή.

## Υλοποίηση

Πρέπει να υποβάλετε ένα αρχείο το οποίο να υλοποιεί τις διαδικασίες `init()`, `changeH()` and `changeV()` και την συνάρτηση `escape()`, as follows:

### Η διαδικασία: `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);
```

### Περιγραφή

Η διαδικασία αυτή σας δίνει την αρχική διάταξη του χάρτη και σας επιτρέπει να αρχικοποιήσετε τις global μεταβλητές και δομές δεδομένων. Θα κληθεί μια μόνο φορά, πριν από οποιοδήποτε κλήσεις στις `changeH()`, `changeV()` ή `escape()`.

### Παράμετροι

- $R$ : ο αριθμός των οριζοντίων δρόμων.
- $C$ : ο αριθμός των κάθετων δρόμων.
- $H$ : ένας δυδιάστατος πίνακας μεγέθους  $R \times (C - 1)$ , όπου  $H[P][Q]$  δίνει τον αριθμό των τεράτων στο τμήμα του οριζόντιου δρόμου μεταξύ των διασταυρώσεων  $(P, Q)$  και  $(P, Q + 1)$ .

- $V$ : ένας δυδιάστατος πίνακας μεγέθους  $(R - 1) \times C$ , όπου  $V[P][Q]$  δίνει τον αριθμό των τεράτων στο τμήμα του κάθετου δρόμου μεταξύ των διασταυρώσεων  $(P, Q)$  και  $(P + 1, Q)$ .

### Η διαδικασία: `init()`

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

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

#### Περιγραφή

Η διαδικασία αυτή καλείται για να αλλάξει τον αριθμό των τεράτων στο τμήμα του οριζόντιου δρόμου μεταξύ των διασταυρώσεων  $(P, Q)$  και  $(P, Q + 1)$ .

#### Παράμετροι

- $P$ : υποδηλώνει ποιός οριζόντιος δρόμος επηρεάζεται ( $0 \leq P \leq R - 1$ ).
- $Q$ : υποδηλώνει ανάμεσα σε ποιούς δύο κάθετους δρόμους βρίσκεται το τμήμα ( $0 \leq Q \leq C - 2$ ).
- $W$ : ο νέος αριθμός τεράτων σε αυτό το τμήμα του δρόμου ( $0 \leq W \leq 1,000$ ).

### Η διαδικασία: `init()`

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

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

#### Περιγραφή

Η διαδικασία αυτή καλείται όταν αλλάζει ο αριθμός των τεράτων σε ένα κάθετο τμήμα δρόμου, ανάμεσα στις διασταυρώσεις  $(P, Q)$  and  $(P + 1, Q)$ .

#### Παράμετροι

- $P$ : υποδηλώνει ποιός οριζόντιος δρόμος επηρεάζεται ( $0 \leq P \leq R - 1$ ).
- $Q$ : υποδηλώνει ανάμεσα σε ποιούς δύο κάθετους δρόμους βρίσκεται το τμήμα ( $0 \leq Q \leq C - 2$ ).
- $W$ : ο νέος αριθμός τεράτων σε αυτό το τμήμα του δρόμου ( $0 \leq W \leq 1,000$ ).

### Η συνάρτησή σας: `escape()`

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

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

## Περιγραφή

Αυτή η συνάρτηση θα πρέπει να υπολογίζει τον μικρότρο δυνατό αριθμό τεράτων τα οποία θα πρέπει να συναντήσει καθώς κινείται από τη διασταύρωση  $(0, V1)$  to  $(R-1, V2)$ .

## Παράμετροι

- `v1` : Προσδιορίζει από που ξεκινά το άτομο στην οριζόντια σειρά  $0$  ( $0 \leq V1 \leq C-1$ ).
- `v2` : Προσδιορίζει που τερματίζει το άτομο στην οριζόντια σειρά  $R-1$  ( $0 \leq V2 \leq C-1$ ).
- *Επιστρέφει*: Τον μικρότερο δυνατό αριθμό τεράτων που είναι υποχρεωμένο το άτομο να συναντήσει.

## Δείγμα

Το παρακάτω περιγράφει το παραπάνω παράδειγμα:

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

## Περιορισμοί

- Χρονικό όριο: 20 δευτερόλεπτα
- Όριο μνήμης: 256 MB(Megabyte)
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Μέγιστος αριθμός αλλαγών (κλήσεις στη `changeH()` ή `changeV()`)
- Το πολύ 200000 κλήσεις στην `escape()`

- Το πολύ 1000 τέρατα ανά τμήμα σε κάθε δεδομένη στιγμή.

## Υποπροβλήματα

Subtask	Points	Additional Input Constraints
1	9	$C = 1$
2	12	$R, C \leq 20$ , και δεν θα υπάρχουν κλήσεις <code>changeH()</code> ή <code>changeV()</code>
3	16	$R, C \leq 100$ , και θα υπάρχουν το πολύ 100 κλήσεις <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(None)

## Πειραματισμός

Το πρόγραμμα grader στον υπολογιστή θα διαβάσει εισόδο από το αρχείο `wombats.in`, το οποίο πρέπει να είναι της ακόλουθης μορφής:

- γραμμή 1: `R C`
- γραμμή 2: `H[0][0] ... H[0][C-2]`
- ...
- γραμμή  $(R + 1)$ : `H[R-1][0] ... H[R-1][C-2]`
- γραμμή  $(R + 2)$ : `V[0][0] ... V[0][C-1]`
- ...
- γραμμή  $(2R)$ : `V[R-2][0] ... V[R-2][C-1]`
- επόμενη γραμμή: `E`
- επόμενες `E` γραμμές: ένα γεγονός ανά γραμμή με την σειρά που αυτά προκύπτουν.

Αν  $C = 1$ , οι άδειες γραμμές που περιέχουν το πλήθος των τεράτων στους οριζόντιους δρόμους (γραμμές 2 μέχρι  $R + 1$ ) δεν είναι απαραίτητες.

Η γραμμή για κάθε γεγονός πρέπει να είναι σε μια από τις παρακάτω μορφές:

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

Για παράδειγμα, για το παραπάνω παράδειγμα τα δεδομένα θα πρέπει να δίδονται στην παρακάτω μορφή:

```
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
```

---

## Σημειώσεις για την γλώσσα προγραμματισμού

- C/C++** Πρέπει να συμπεριλάβετε την οδηγία `#include "wombats.h"`.
- Pascal** Πρέπει να ορίσετε `unit Wombats`. Σε όλους τους πίνακες η αρίθμηση των στοιχείων αρχίζει από `0` (όχι `1`).

Δείτε τις προτεινόμενες λύσεις στον υπολογιστή σας για παραδείγματα