

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 22/8/2014, 14:59:59

Νυχτερίδες και αράχνες... (0.25+0.25 = 0.5 βαθμοί)

Στην πρόσφατη κατάληψη του ΕΜΠ, οι νυχτερίδες και οι αράχνες που ζουν στα κτίρια ένωσαν πολύ μόνες. Για να μη βαρεθούν εντελώς αποφάσισαν να ξεσκονίσουν τις γνώσεις τους στην Ευκλείδεια γεωμετρία με το να υπολογίζουν μήκη ευθύγραμμων τμημάτων. Κάθε μέρα μία και μόνο αράχνη διάλεγε κάποια θέση σε ένα κτίριο και κάποιες από τις νυχτερίδες διάλεγαν επίσης θέσεις σε κάποια άλλα σημεία του κτιρίου. Αν θεωρήσουμε ότι τα κτίρια είναι ορθογώνια, τότε ο χάρτης ενός κτιρίου μπορεί να περιγραφεί από τις διαστάσεις **N** και **M** του κτιρίου και τον αριθμό **K** των συντεταγμένων στο χάρτη όπου υπάρχει κάτι ενδιαφέρον: είτε τοίχος (συμβολίζεται με τον χαρακτήρα “-”), είτε νυχτερίδα (συμβολίζεται με τον ASCII χαρακτήρα “B”), είτε αράχνη (συμβολίζεται με το ASCII “A”). Έχουμε δηλαδή θεωρήσει ότι το κτίριο αποτελείται από $N \times M$ τετράγωνα, οι συντεταγμένες των οποίων είναι από (0, 0) έως (N-1, M-1).

Ας θεωρήσουμε ότι η απόσταση μεταξύ των κέντρων γειτονικών τετραγώνων (είτε οριζόντια ή κάθετα) είναι 1, και ότι η απόσταση όλων των άλλων κέντρων τετραγώνων υπολογίζεται με χρήση Ευκλείδειας γεωμετρίας. Αυτό που θέλουμε να κάνουμε είναι να υπολογίσουμε την ελάχιστη απόσταση της τεθλασμένης γραμμής που ενώνει κέντρα τετραγώνων του χάρτη ως εξής: ξεκινά από το τετράγωνο με συντεταγμένες (0,0), στο οποίο πάντα βρίσκεται μια νυχτερίδα, καταλήγει στο τετράγωνο στο οποίο βρίσκεται η αράχνη, σε κάθε τετράγωνο που βρίσκεται σε ενδιάμεση κορυφή της τεθλασμένης γραμμής υπάρχει κάποια νυχτερίδα, και η γραμμή δε συγκρούεται (εφάπτεται ή τέμνεται) πουθενά με κάποιον τοίχο.

Η άσκηση ζητάει να γραφούν δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία παίρνουν ως είσοδο το χάρτη και επιστρέφουν ως έξοδο έναν αριθμό με το πολύ δύο δεκαδικά ψηφία: την ελάχιστη απόσταση από τη νυχτερίδα στο αρχικό τετράγωνο μέχρι την αράχνη.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου θα περιέχει τους τρεις φυσικούς αριθμούς **N**, **M** και **K**. Οι επόμενες **K** γραμμές θα περιέχουν τις συντεταγμένες στις οποίες υπάρχει κάτι ενδιαφέρον και τι είναι αυτό: ένας από τους χαρακτήρες “-”, “B” ή “A”. Θεωρήστε επίσης δεδομένο ότι στο τετράγωνο με συντεταγμένες (0, 0) υπάρχει πάντα ένα “B” και ότι θα υπάρχει πάντοτε τεθλασμένη γραμμή που να πληροί τις παραπάνω προδιαγραφές.

Περιορισμοί: $3 \leq N, M \leq 1000$, όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

Παρακάτω δείχνουμε κάποιες πιθανές κλήσεις των προγραμμάτων σε ML και σε Java.

Σε SML/NJ

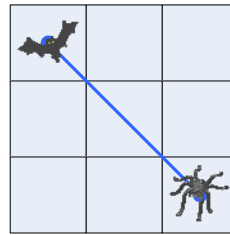
```
- bats "bats1.txt";  
val it = 2.83 : real  
  
- bats "bats2.txt";  
val it = 4.0 : real  
  
- bats "bats3.txt";  
val it = 9.24 : real
```

Σε Java

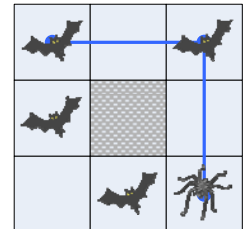
```
> java Bats bats1.txt  
2.83  
  
> java Bats bats2.txt  
4.0  
  
> java Bats bats3.txt  
9.24
```

όπου τα αρχεία με τους χάρτες είναι τα εξής (η εντολή `cat` είναι εντολή του Unix):

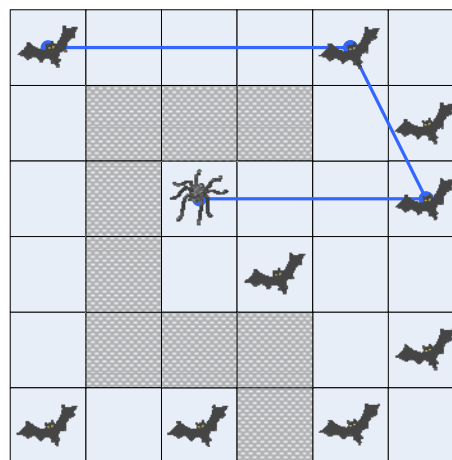
```
> cat bats1.txt
3 3 2
0 0 B
2 2 A
```



```
> cat bats2.txt
3 3 6
0 0 B
0 2 B
1 0 B
1 1 -
2 1 B
2 2 A
```



```
> cat bats3.txt
6 6 19
0 0 B
0 4 B
1 1 -
1 2 -
1 3 -
1 5 B
2 1 -
2 2 A
2 5 B
3 1 -
3 3 B
4 1 -
4 2 -
4 3 -
4 5 -
5 0 B
5 2 B
5 3 -
5 4 B
```



Στο πρώτο παράδειγμα η λύση είναι προφανής: η ελάχιστη απόσταση είναι η διαγώνιος με μήκος $\sqrt{8} = 2.82842\dots$ το οποίο με στρογγυλοποίηση σε δύο δεκαδικά ψηφία γίνεται 2.83. Στο δεύτερο παράδειγμα η λύση αποτελείται από δύο ευθύγραμμα τμήματα που ενώνουν τη νυχτερίδα στο τετράγωνο (0,0) με αυτή στο (0,2) και μετά αυτή στο (0,2) με την αράχνη στο (2,2). Τα τρία ευθύγραμμα τμήματα (0,0) – (1,0), (1,0) – (2,1) και (2,1) – (2,2) είναι μεν μικρότερα ως συνολική απόσταση, αλλά δεν αποτελούν λύση διότι το τμήμα από το (1,0) στο (2,1) εφάπτεται σε μια γωνία του τετραγώνου (1,1) όπου υπάρχει τοίχος. Στο τρίτο παράδειγμα η λύση αποτελείται από τα τρία ευθύγραμμα τμήματα (0,0) – (0,4), (0,4) – (2,5) και (2,5) – (2,2).

Κουβαδάκια στην παραλία... (0.25+0.25 = 0.5 βαθμοί)

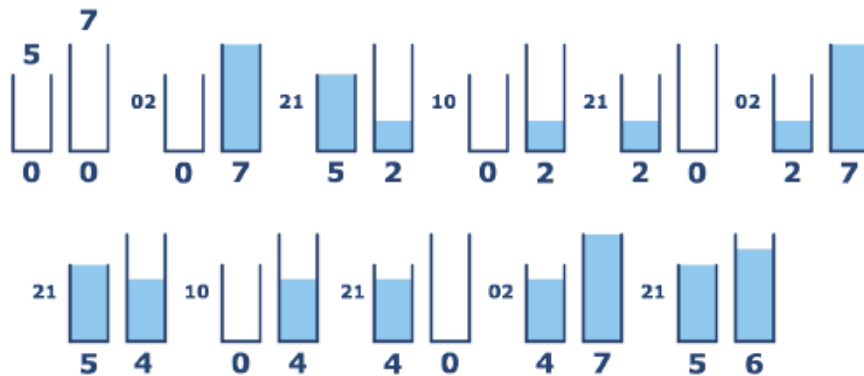
Η κόρη μου είναι στην ηλικία που παίζει με κουβαδάκια στην παραλία. Επειδή από και προς την παραλία τα κουβαδάκια τα κουβαλάω εγώ, της έχουμε αγοράσει μόνο δύο. Της αρέσει πολύ να γεμίζει κάποιο κουβαδάκι της με νερό από τη θάλασσα, να αδειάζει το περιεχόμενο του ενός στο άλλο, ή να αδειάζει κάποιο κουβαδάκι όλο πίσω στη θάλασσα. Για να κάνω το παιχνίδι της λίγο πιο ενδιαφέρον για μένα, την ώρα που παίζουμε με τα κουβαδάκια εγώ στο μυαλό μου λύνω το εξής πρόβλημα. Έστω ότι τα δύο κουβαδάκια έχουν χωρητικότητες V_1 και V_2 (ας πούμε σε λίτρα) και έστω ότι θέλω τελικά να έχω ακριβώς V_g λίτρα νερού σε ένα από τα δύο κουβαδάκια. Το ζητούμενο είναι με ποια ελάχιστη ακολουθία κινήσεων μπορώ να φτάσω στο στόχο, όπου οι επιτρεπτές «κινήσεις» είναι οι εξής:

- $0k$: να γεμίσω από τη θάλασσα το κουβαδάκι k (όπου $k \in \{1,2\}$)

- $k0$: να αδειάσω το κουβαδάκι k στη θάλασσα (όπου $k \in \{1,2\}$)
- kk' : να αδειάσω το κουβαδάκι k στο κουβαδάκι k' (όπου $k, k' \in \{1,2\}$ και τα k και k' είναι διαφορετικά μεταξύ τους). Όταν ένα κουβαδάκι αδειάζει σε ένα άλλο, είτε αδειάζει όλο το περιεχόμενό του ή αδειάζει μέχρι το σημείο στο οποίο γεμίζει πλήρως το άλλο κουβαδάκι.

Ας δούμε πρώτα ένα απλό παράδειγμα. Έστω ότι τα κουβαδάκια έχουν χωρητικότητες $V1=3$ και $V2=4$ και ο στόχος είναι $Vg=1$. Μπορούμε να γεμίσουμε το δεύτερο κουβαδάκι από τη θάλασσα (02) και να αδειάσουμε το περιεχόμενό του στο πρώτο (21) μέχρι να γεμίσει και να αφήσει όγκο 1 στο δεύτερο κουβαδάκι. Η λύση λοιπόν είναι 02-21.

Για ένα πιο πολύπλοκο παράδειγμα, έστω ότι τα κουβαδάκια έχουν χωρητικότητες $V1=5$ και $V2=7$ και ο στόχος είναι $Vg=6$. Η λύση είναι 02-21-10-21-02-21-10-21-02-21 και φαίνεται παρακάτω.



Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία να παίρνουν ως είσοδο τα $V1$, $V2$ και Vg και να επιστρέφουν ως έξοδο μια συμβολοσειρά με τις ελάχιστες κινήσεις για να επιτευχθεί ο επιθυμητός όγκος. Αν δεν είναι δυνατό να επιτευχθεί ο στόχος, το πρόγραμμα πρέπει να επιστρέφει τη συμβολοσειρά "impossible".

Περιορισμοί: χωρητικότητες ≤ 42 , όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

Στην άσκηση αυτή η είσοδος είναι πολύ απλή και δεν έχει νόημα να είναι σε αρχείο. Παρακάτω δείχνουμε κάποιες πιθανές κλήσεις των προγραμμάτων σε ML και σε Java.

Σε SML/NJ

```
- kouvadakia 3 4 1;
val it = "02-21" : string
- kouvadakia 5 7 6;
val it = "02-21-10-21-02-21-10-21-02-21" : string
- kouvadakia 4 2 1;
val it = "impossible" : string
```

Σε Java

```
> java Kouvadakia 3 4 1
02-21
> java Kouvadakia 5 7 6
02-21-10-21-02-21-10-21-02-21
> java Kouvadakia 4 2 1
impossible
```

Περαιτέρω οδηγίες για την άσκηση

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με την προηγούμενη σειρά ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά ασκήσεων.
- Δεν επιτρέπεται να μοιράζετε τα προγράμματά σας με συμφοιτητές εκτός της ομάδας σας ή να τα βάλετε σε μέρος που άλλοι μπορούν να τα βρουν (π.χ. σε κάποια σελίδα στο διαδίκτυο,

σε ιστοσελίδες συζητήσεων, ...). Σε περίπτωση που παρατηρηθούν «περίεργες» ομοιότητες σε προγράμματα, ο βαθμός των εμπλεκόμενων φοιτητών σε *όλες τις σειρές ασκήσεων* γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.

- Τα προγράμματα σε ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ v110.74 ή σε MLton 20100608 ή σε Objective Caml version 3.12.1. Το σύστημα ηλεκτρονικής υποβολής επιτρέπει να επιλέξετε μεταξύ αυτών των υλοποιήσεων της ML.
- Ο κώδικας των προγραμμάτων σε Java μπορεί να βρίσκεται σε περισσότερα του ενός αρχείου αν θέλετε αλλά θα πρέπει να μπορεί να μεταγλωττιστεί χωρίς προβλήματα με τον Java compiler με εντολές της μορφής `javac Bats.java` ή `javac Kounadakia.java`.
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle, όπως και στην προηγούμενη άσκηση, και για να μπορέσετε να τις υποβάλλετε, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχουν ήδη λογαριασμό στο moodle. Θα υπάρξει σχετική ανακοίνωση μόλις το σύστημα υποβολής καταστεί ενεργό. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλου είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.