

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 7/5/2017, 23:59:59

Εκδρομή για σκι (0.25+0.25 = 0.5 βαθμοί)

Η Κατερίνα θέλει να πάει για σκι. Το χιονοδρομικό κέντρο είναι μακριά από το σπίτι της, κι έτσι παίρνει τα πέδιλα στον ώμο και αποφασίζει να πάει στο βουνό με τον οδοντωτό σιδηρόδρομο. Ο σιδηρόδρομος κάνει στάσεις σε N σημεία στο βουνό. Οποιοσδήποτε δύο διαδοχικές στάσεις απέχουν μεταξύ τους ένα χιλιόμετρο. Για κάθε στάση i (όπου $1 \leq i \leq N$) γνωρίζουμε το υψόμετρο Y_i στο οποίο αυτή βρίσκεται.



spoiler!

Καθώς ο σιδηρόδρομος απομακρύνεται στο βουνό, η Κατερίνα μπορεί να κατέβει το πρώι σε όποια στάση θέλει, να κάνει σκι με κατεύθυνση προς τα πίσω και να επιβιβαστεί πάλι στο σιδηρόδρομο σε κάποια προηγούμενη στάση, όταν αυτός θα επιστρέφει το απόγευμα, ώστε να γυρίσει στο σπίτι της. Όμως, για να μπορέσει να κάνει σκι, η στάση που θα κατέβει πρέπει να μην είναι σε χαμηλότερο υψόμετρο από εκείνην που θα επιβιβαστεί και πάλι. Στην Κατερίνα αρέσει πολύ το σκι. Βοηθήστε τη να βρει τη μεγαλύτερη διαδρομή που μπορεί να κάνει.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν τις τιμές των N και Y_i και να επιστρέφουν ως έξοδο το μήκος της μεγαλύτερης διαδρομής που μπορεί να κάνει η Κατερίνα.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει τον ακέραιο N ($2 \leq N \leq 2.000.000$). Η δεύτερη γραμμή περιέχει την ακολουθία με τα N υψόμετρα Y_i . Τα υψόμετρα είναι θετικοί ακέραιοι το πολύ μέχρι 1.000.000.000.

Παρακάτω δίνονται κάποια παραδείγματα σε C/C++ και σε ML.

Σε C/C++, MLton, ή σε OCaml

```
$ ./skitrip h1.txt  
10
```

```
$ ./skitrip h2.txt  
0
```

Σε SML/NJ

```
- skitrip "h1.txt";  
val it = 10 : int
```

```
- skitrip "h2.txt";  
val it = 0 : int
```

όπου τα αρχεία με τα δεδομένα εισόδου είναι τα εξής (η εντολή `cat` είναι εντολή του Unix):

```
$ cat h1.txt  
16  
78 88 64 94 17 91 57 69 38 62 13 17 35 15 20 15  
  
$ cat h2.txt  
6  
42 41 40 39 38 37
```

Για το πρώτο αρχείο εισόδου, η Κατερίνα μπορεί να κατέβει στη 15η στάση, που έχει υψόμετρο $Y_{15} = 20$. Από εκεί, μπορεί να κάνει σκι προς τα πίσω μέχρι την 5η στάση, που έχει υψόμετρο $Y_5 = 17$. Η διαδρομή αυτή είναι η μεγαλύτερη που μπορεί να κάνει και έχει μήκος $15 - 5 = 10$ χιλιόμετρα.

Για το δεύτερο, τα υψόμετρα των στάσεων μειώνονται συνεχώς και δεν υπάρχει έγκυρη διαδρομή προς τα πίσω. Η Κατερίνα είναι άτυχη και θα πρέπει να περιμένει στην ίδια στάση που θα κατέβει, χωρίς να κάνει σκι, μέχρι να γυρίσει ο σιδηρόδρομος το απόγευμα.

Διαστημικές παραγγελίες (0.25+0.25 = 0.5 βαθμοί)

Δίνεται ένας διδιάστατος χάρτης του διαστήματος, αποτελούμενος από $N \times M$ τετραγωνάκια ($4 \leq N, M \leq 1000$). Βοηθήστε τον Λάκη τον εξωγήινο, που δουλεύει ντελιβεράς στο εστιατόριο στην άκρη του σύμπαντος, να παραδώσει την παραγγελία με τον ιπτάμενο δίσκο του το νωρίτερο δυνατό.



spoiler safe

Κάθε τετραγωνάκι του χάρτη περιέχει ένα από τα εξής σύμβολα:

- “S” Η αρχική θέση του Λάκη
- “E” Η τελική θέση, δηλαδή ο προορισμός της παραγγελίας
- “X” Εμπόδιο, δηλαδή τετράγωνο όπου ο Λάκης δεν μπορεί — ή δεν θέλει — να βρεθεί (θέσεις άστρων, μαύρες τρύπες, ο Νταρθ Βέηντερ αυτοπροσώπως, κ.λπ.)
- “W” Σκουληκότρυπα (wormhole)
- “.” (τελεία): Κενό — τι άλλο περιμένετε να βρείτε στο διάστημα;

Ο Λάκης μπορεί να ταξιδεύει κινούμενος μεταξύ γειτονικών τετραγώνων στο χάρτη, δεν μπορεί όμως να πάει σε τετράγωνο με εμπόδιο. Μπορεί να χρησιμοποιήσει το δίκτυο των σκουληκότρυπων, οι οποίες συνδέονται μεταξύ τους και από οποιαδήποτε μπορεί να τηλεμεταφερθεί οπουδήποτε. Όμως, ο Λάκης φοβάται να τηλεμεταφερθεί ο ίδιος, γιατί κάποτε είχε ακούσει ότι ένας ιπτάμενος δίσκος που μπήκε σε μία σκουληκότρυπα κατέληξε στην Ελλάδα του 2018, παραμονές εκλογών. Δε διστάζει όμως να χρησιμοποιήσει τις σκουληκότρυπες (όσες χρειαστεί) για να τηλεμεταφέρει την παραγγελία του.

Οι κινήσεις που μπορεί να κάνει ο Λάκης παριστάνονται με τα σύμβολα:

- “R” Κίνηση ένα τετράγωνο δεξιά στο χάρτη
- “L” Κίνηση ένα τετράγωνο αριστερά στο χάρτη
- “U” Κίνηση ένα τετράγωνο προς τα πάνω στο χάρτη
- “D” Κίνηση ένα τετράγωνο προς τα κάτω στο χάρτη
- “W” Τοποθέτηση της παραγγελίας σε μία σκουληκότρυπα ή παραλαβή της παραγγελίας από μία σκουληκότρυπα

Ο Λάκης ξεκινάει από την αρχική του θέση (με την παραγγελία φορτωμένη) και πρέπει να πάει στην τελική θέση (μαζί με την παραγγελία προφανώς), χρησιμοποιώντας τέτοιες κινήσεις. Μπορεί να κινείται αριστερά, δεξιά, πάνω και κάτω, και κάθε τέτοια κίνηση του κοστίζει 2 λεπτά, αν μεταφέρει την παραγγελία, ή 1 λεπτό, αν δεν την μεταφέρει. (Αν αναρρωτιέστε γιατί συμβαίνει αυτό, η διαστημική παραγγελία που μεταφέρει ο Λάκης αλληλεπιδρά με το warp drive που δουλεύει στη μισή ταχύτητα όταν του μυρίζει φαγητό.) Επίσης, όταν βρίσκεται στο τετράγωνο μίας σκουληκότρυπας, μπορεί να τοποθετήσει εκεί την παραγγελία του και (αργότερα) να την παραλάβει από οποιαδήποτε άλλη σκουληκότρυπα. Και τα δύο αυτά είδη κινήσεων κοστίζουν στον Λάκη 1 λεπτό.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία βρίσκουν μία ελάχιστου κόστους ακολουθία κινήσεων, την οποία μπορεί να χρησιμοποιήσει ο Λάκης για να παραδώσει την παραγγελία του. Αν υπάρχουν περισσότερες διαφορετικές ακολουθίες με ελάχιστο κόστος, αρκεί να βρείτε μία (οποιαδήποτε) από αυτές.

Η είσοδος του προγράμματός σας διαβάζεται από ένα αρχείο αποτελούμενο από N γραμμές, κάθε μία από τις οποίες περιέχει M σύμβολα. Το αρχείο αυτό αναπαριστά το χάρτη.

Παρακάτω δίνονται κάποια παραδείγματα σε C/C++ και σε ML.

Σε C/C++, MLton, ή σε OCaml

```
$ ./spacedeli map1.txt
12 RRRDDD

$ ./spacedeli map2.txt
11 DDWRRRWD
```

Σε SML/NJ

```
- spacedeli "map1.txt";
val it = (12,"RRRDDD") : int * string

- spacedeli "map2.txt";
val it = (11,"DDWRRRWD") : int * string
```

όπου τα αρχεία εισόδου είναι τα εξής (η εντολή `cat` είναι εντολή του Unix και από μόνη της δεν συνηθίζει να χρωματίζει χάρτες στο διάστημα):

```
$ cat map1.txt
S . . .
. . .
W . XW
. . XE

$ cat map2.txt
S . . .
. . .
W . . W
. . XE
```

Στο πρώτο παράδειγμα, ο Λάκης μπορεί να πάει με έξι κινήσεις κατευθείαν από την αρχική του θέση στον προορισμό της παραγγελίας, με κόστος $6 \times 2 = 12$. Δυστυχώς δεν υπάρχει μικρότερου κόστους διαδρομή.

Στο δεύτερο παράδειγμα, ο Λάκης μπορεί να κατέβει στην αριστερή σκουληκότρυπα (κόστος $2 \times 2 = 4$), να τοποθετήσει εκεί την παραγγελία που τηλεμεταφέρεται στη δεξιά σκουληκότρυπα (κόστος 1), στη συνέχεια να πάει στη δεξιά σκουληκότρυπα (κόστος $3 \times 1 = 3$), να παραλάβει την παραγγελία (κόστος 1), και να πάει ένα τετράγωνο κάτω για να την παραδώσει (κόστος 1×2). Συνολικό κόστος $4 + 1 + 3 + 1 + 2 = 11$.

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων, τόσο σε αυτή όσο και στις επόμενες σειρές ασκήσεων. Όμως, έχετε υπ' όψη σας ότι, αν δεν περάσετε το μάθημα φέτος, οι βαθμοί των προγραμματιστικών ασκήσεων κρατούνται μόνο για όσους δεν τις έκαναν σε ομάδα αλλά τις έκαναν μόνοι τους.
- Δεν επιτρέπεται να μοιράζεστε 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.
- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. κώδικα ταξινόμησης, κάποιο κώδικα που διαχειρίζεται κάποια δομή δεδομένων) που βρήκατε στο διαδίκτυο στα προγράμματά σας, με την προϋπόθεση ότι το πρόγραμμά σας περιέχει σε σχόλια την παραδοχή για την προέλευση αυτού του κώδικα και ένα σύνδεσμο σε αυτόν.
- Τα προγράμματα σε C/C++ πρέπει να είναι σε ένα αρχείο και να μπορούν να μεταγλωττιστούν χωρίς warnings με `gcc/g++ (version ≥ 4.9.2)` με εντολές της μορφής, π.χ.

```
gcc -std=c99 -Wall -Werror -O3 -o skitrip skitrip.c
g++ -std=c++11 -Wall -Werror -O3 -o skitrip skitrip.cpp
```

- Τα προγράμματα σε ML πρέπει επίσης να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ $\geq v110.76$ ή σε MLton ≥ 20100608 ή σε Objective Caml version $\geq 4.01.0$. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.
- Η αποστολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του moodle 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 για την ακριβή διαδικασία υποβολής όταν ανοίξει το σύστημα. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλου είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.