

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 6/6/2013, 23:59:59

Τάσο μου τη Γκόλφω σου... (0.25+0.25 = 0.5 βαθμοί)

Δίνεται ένας χάρτης, αποτελούμενος από $N \times N$ τετραγωνάκια. Κάθε τετραγωνάκι μπορεί να είναι κενό (συμβολίζεται με ένα κενό διάστημα) ή να περιέχει εμπόδιο (συμβολίζεται με το γράμμα "X"). Επίσης, σε ένα τετραγωνάκι βρίσκεται ο Τάσος (συμβολίζεται με το γράμμα "T") και σε ένα άλλο η Γκόλφω (συμβολίζεται με το γράμμα "G").

Ο Τάσος επιθυμεί να συναντηθεί με τη Γκόλφω. Κάθε φορά, μπορεί να κινηθεί σε ένα από τα γειτονικά τετραγωνάκια (πάνω, κάτω, αριστερά και δεξιά). Η κίνηση προς τα αριστερά ή τα δεξιά κοστίζει 1 μονάδα, η κίνηση προς τα πάνω κοστίζει 2 μονάδες, ενώ η κίνηση προς τα κάτω (γνωστή και ως «πέσιμο») δεν κοστίζει (παρά μόνο σε έξοδα νοσοκομείου).

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία να παίρνουν ως είσοδο το χάρτη και να επιστρέφουν ως έξοδο έναν αριθμό: το ελάχιστο κόστος για τον Τάσο μέχρι να φτάσει στην Γκόλφω. Σε περίπτωση που ο Τάσος δεν είναι δυνατόν να φτάσει στην Γκόλφω, τα προγράμματα θα επιστρέφουν την τιμή -1 (~1 στην SML).

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου θα περιέχει ένα φυσικό αριθμό N : τη διάσταση του χάρτη. Οι επόμενες N γραμμές θα περιέχουν το χάρτη. Κάθε μία θα περιέχει N χαρακτήρες, καθένας από τους οποίους θα είναι είτε ένα κενό διάστημα, είτε ένα από τα γράμματα "X", "T" ή "G". Θα υπάρχει ακριβώς ένα "T" και ακριβώς ένα "G". Θεωρήστε επίσης δεδομένο ότι οι χαρακτήρες που βρίσκονται στα όρια του χάρτη θα είναι πάντοτε "X".

Περιορισμοί: $4 \leq N \leq 500$, όριο χρόνου εκτέλεσης: 10 seconds, όριο μνήμης: 256 MB.

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

Σε SML/NJ	Σε Java
- golfo "golfo1.txt";	> java Golfo golfo1.txt
val it = 20 : int	20
- golfo "golfo2.txt";	> java Golfo golfo2.txt
val it = ~1 : int	-1

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

```
> cat golfo1.txt
10
XXXXXXXXXX
X          X
X T        X
XXXXXXXXXX X
X          X X
X X        X
X XXXXXXXX
X          GX
X          X
XXXXXXXXXXXX
```

```
> cat golfo2.txt
8
XXXXXXXXX
X T      X
XXXXXXX X
X  X  X
X  XXXXX
X      X
X      GX
XXXXXXXXX
```

Ο Τάσος και τα πρόβατα... (0.25+0.25 = 0.5 βαθμοί)

Δυστυχώς, λόγω κρίσης, ο Τάσος δε μπορεί απλώς να συναντιέται όλη μέρα με τη Γκόλφω, αλλά πρέπει να βγάζει και λεφτά για να μπορεί να πληρώνει την εφορία, τα χαράτσια, κ.λπ. Η δουλειά του είναι να «τακτοποιεί» πρόβατα στις στάνες του χωριού του: κάθε πρωί ξεκινάει από το σπίτι του με μία λίστα όλων των προβάτων που πρέπει να μετακινηθούν από μια στάνη σε κάποια άλλη και θέλει να ολοκληρώσει τη δουλειά με την ελάχιστη δυνατή προσπάθεια. Λόγω της ακρίβειας της βενζίνης, για τις μετακινήσεις χρησιμοποιεί τα πόδια του και φορτώνει τα πρόβατα στην πλάτη του, το ένα πάνω στο άλλο αν χρειαστεί. Κατά συνέπεια, όταν θέλει να ξεφορτώσει ένα συγκεκριμένο πρόβατο σε κάποια στάνη πρέπει πρώτα να ξεφορτώσει και όλα τα άλλα που βρίσκονται πάνω από αυτό. (Με άλλα λόγια η πλάτη του Τάσου είναι... στοιβαρή!) Οι «κινήσεις» που μπορεί να κάνει ο Τάσος είναι:

- **move k**: να μετακινηθεί από το σημείο που βρίσκεται στη στάνη **k**, με κόστος 3 μονάδες
- **load p**: να φορτώσει το πρόβατο **p** από τη στάνη στην οποία βρίσκεται (τόσο ο Τάσος όσο και το πρόβατο) στην πλάτη του, πάνω από όλα τα άλλα πρόβατα που τυχόν βρίσκονται εκεί, με κόστος 1 μονάδα
- **unload p**: να ξεφορτώσει το πρόβατο **p** που έχει πάνω-πάνω στην πλάτη του στη στάνη στην οποία βρίσκεται ο Τάσος τη συγκεκριμένη χρονική στιγμή, με κόστος 1 μονάδα
- **flute**: να μην κάνει τίποτε από τα παραπάνω και απλώς να... παίξει τη φλογέρα του, με κόστος 0 μονάδες. :-)

Αυτό που ζητάει η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε ML και ένα σε Java) τα οποία να παίρνουν ως είσοδο τη λίστα με τα πρόβατα που πρέπει να μετακινηθούν και να επιστρέφουν ως έξοδο έναν αριθμό: το ελάχιστο κόστος για να μετακινηθούν όλα στη θέση τους. Υποθέστε ότι ο Τάσος αρχικά είναι στο σπίτι του με άδεια πλάτη και πρέπει να μετακινηθεί στη στάνη στην οποία βρίσκεται κάποιο πρόβατο, αν θέλει να το πάρει στην πλάτη του.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο, όπως φαίνεται στα παραδείγματα που ακολουθούν. Η πρώτη γραμμή του αρχείου περιέχει δύο φυσικούς αριθμούς **K** και **N**: το πόσες στάνες υπάρχουν και τον αριθμό των προβάτων που θα μετακινηθούν. Οι επόμενες **N** γραμμές περιέχουν πληροφορία για τις μετακινήσεις. Κάθε μία περιέχει δύο αριθμούς **S** και **D**: τη στάνη στην οποία βρίσκεται αρχικά το πρόβατο και αυτή στην οποία πρέπει να μετακινηθεί. Μπορείτε να θεωρήσετε δεδομένο ότι τα **S** και **D** είναι διαφορετικά μεταξύ τους.

Περιορισμοί: $0 \leq K \leq 42$, $0 \leq N \leq 42$, όριο μνήμης: 4 GB, όριο χρόνου εκτέλεσης: 2 ώρες.

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

```
Σε SML/NJ
- provata "provata1.txt";
val it = 8 : int
- provata "provata2.txt";
val it = 13 : int
- provata "provata3.txt";
val it = 20 : int
```

```
Σε Java
> java Provata provata1.txt
8
> java Provata provata2.txt
13
> java Provata provata3.txt
20
```

όπου τα τρία αρχεία με τις πληροφορίες για τις μετακινήσεις των προβάτων είναι τα εξής:

```

> cat provata1.txt
2 1
1 2

> cat provata2.txt
3 2
1 2
1 3

> cat provata3.txt
3 4
1 2
1 3
2 1
3 2

```

Για το πρώτο αρχείο εισόδου η λύση είναι προφανής: ο Τάσος πρέπει να μετακινηθεί από το σπίτι του στη στάνη 1, να φορτώσει το πρόβατο στην πλάτη του, να μετακινηθεί στη στάνη 2 και να το ξεφορτώσει με συνολικό κόστος $3+1+3+1=8$. Ενδιάμεσα, μπορεί να παίξει τη φλογέρα του όσες φορές θέλει αλλά αυτό δεν αλλάζει το κόστος.

Χωρίς μουσικά διαλείμματα, το δεύτερο αρχείο εισόδου έχει δύο λύσεις με ελάχιστο κόστος: $3+1+1+3+1+3+1=13$.

Τέλος, για το τρίτο αρχείο εισόδου, ο Τάσος μπορεί να επιλέξει μεταξύ μιας λύσης ξεκινώντας από τη στάνη 1 με κόστος $3+1+1+3+1+1+3+1+1+1+3+1=20$ και μιας ξεκινώντας από τη στάνη 2 με κόστος $3+1+3+1+1+1+3+1+1+3+1+1=20$. Υπάρχουν και ακολουθίες κινήσεων με μεγαλύτερο κόστος (π.χ. ξεκινώντας από τη στάνη 3 με συνολικό κόστος 23) αλλά φυσικά δεν είναι αυτές που ζητάει η άσκηση.

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 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 στις ασκήσεις 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 3.11.2. Το σύστημα ηλεκτρονικής υποβολής επιτρέπει να επιλέξετε μεταξύ αυτών των υλοποιήσεων της ML.
- Ο κώδικας των προγράμματα σε Java μπορεί να βρίσκεται σε περισσότερα του ενός αρχείου αλλά θα πρέπει να μπορεί να μεταγλωττιστεί χωρίς προβλήματα με τον Java compiler με εντολές της μορφής `javac Golfo.java` ή `javac Provata.java`
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά όπως και στην προηγούμενη άσκηση. 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 άλλους είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.