ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εαρινό 2024

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 24/5/2024, 23:59:59

Το Δένδρο του Κόσμου, ξανά (0.25 βαθμοί)

Το πρόβλημα αυτό σας είναι γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Java. Το πρόγραμμά σας μπορεί να είναι σε πολλά αρχεία **`.java`** αν το επιθυμείτε, αλλά η κύρια κλάση του (δηλ. αυτή με τη μέθοδο **`main`**) θα πρέπει να ονομάζεται **`Arrange`**. Η **`main`** σας θα πρέπει να τυπώνει, σε μια γραμμή όπως και στη C/C++, την λεξικογραφικά ελάχιστη ακολουθία που μπορεί να προκύψει από την ενδοδιατεταγμένη διάσχιση του δέντρου, μετά από την εφαρμογή οσωνδήποτε κατάλληλων πράξεων αντιμετάθεσης στους κόμβους του δέντρου. Για παράδειγμα

```
$ java Arrange tree1.txt
1 6 4 2 3 5
```

Για το διάβασμα της εισόδου, δείτε το υπόδειγμα που δίνεται στην εκφώνηση της πρώτης σειράς ασκήσεων.

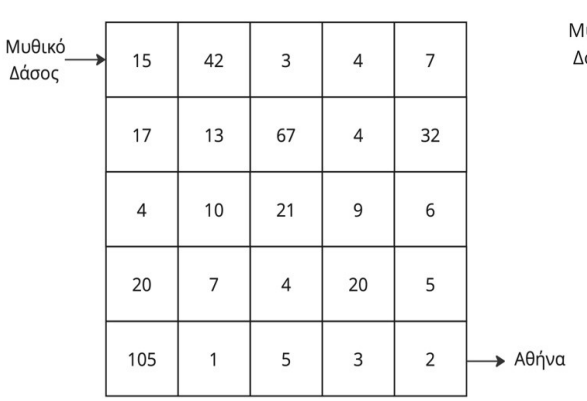
Επιστροφή στην Αθήνα (0.25+0.25 = 0.5 βαθμοί)

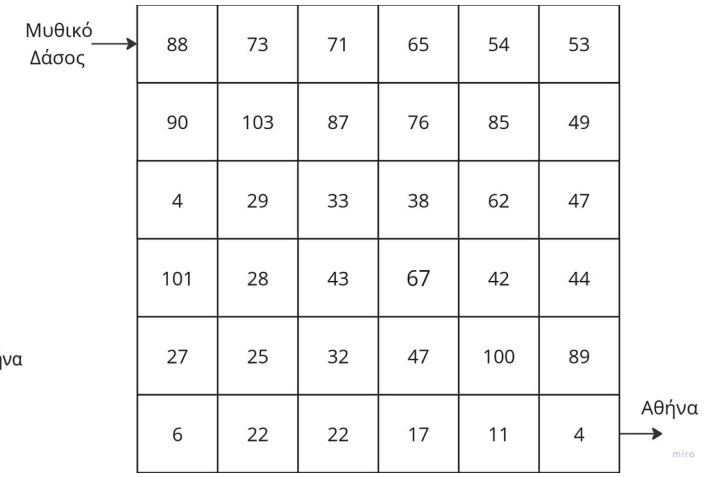
Ο Βαγγέλης επέλυσε με επιτυχία τον γρίφο του Δέντρου του Κόσμου στις διακοπές του Πάσχα στο χωριό του. Τώρα πρέπει να γυρίσει στην Αθήνα. Το ίδιο όμως θέλουν να κάνουν και χιλιάδες άλλοι που εγκατέλειψαν την πρωτεύουσα για να περάσουν το Πάσχα στην εξοχή. Ο κίνηση στους δρόμους όμως είναι μεγάλη και ο Βαγγέλης δεν έχει χρόνο για χάσιμο: πρέπει να γυρίσει όσο το δυνατόν γρηγορότερα για να ασχοληθεί με τις επόμενες σειρές ασκήσεων γλωσσών προγραμματισμού.

Προκειμένου να γυρίσει στην Αθήνα, ο Βαγγέλης πρέπει να διασχίσει ένα πλέγμα μεγέθους NxN. Κάθε τετράγωνο περιέχει έναν ακέραιο που υποδηλώνει τον αριθμό των αυτοκινήτων που βρίσκονται στο τετράγωνο. Σύμφωνα με τον ΚΟΚ επιτρέπεται η μετακίνηση σε όλα τα γειτονικά τετράγωνα, μόνο όμως αν έχουν μικρότερο αριθμό αυτοκινήτων από το τρέχον τετράγωνο. Η διαδρομή ξεκινάει από το πάνω αριστερά τετράγωνο και τελειώνει στο κάτω δεξιά.

Η άσκηση σας ζητάει να γράψετε ένα πρόγραμμα σε Java και ένα σε Prolog που να δέχονται ως είσοδο ένα αρχείο που περιγράφει το πλέγμα και να επιστρέφουν ως έξοδο τη μικρότερη δυνατή ακολουθία κινήσεων που μπορεί να κάνει ο Βαγγέλης για να επιστρέψει στην Αθήνα. Η πρώτη γραμμή του αρχείου εισόδου περιέχει το μέγεθος N του τετραγωνικού πλέγματος. Κάθε μία από τις επόμενες N γραμμές περιέχει N αριθμούς που υποδηλώνουν τον αριθμό των αυτοκινήτων στο αντίστοιχο τετράγωνο. Η διαδρομή είναι μια ακολουθία κινήσεων σε κάθε μια από τις οκτώ δυνατές κατευθύνσεις (**`N`**, **`S`**, **`W`**, **`E`**, **`NE`**, **`NW`**, **`SE`** και **`SW`**). Το πρόγραμμα Java θα πρέπει να τυπώνει την ακολουθία των βημάτων ως ένα string που περιέχει μια λίστα ή το string **`IMPOSSIBLE`** αν δεν υπάρχει δυνατή διαδρομή. Το πρόγραμμα Prolog πρέπει να ορίζει ένα κατηγόρημα **`moves/2`** το οποίο επιτυγχάνει με *μια μόνο* απάντηση (μια λίστα με την ακολουθία των βημάτων, με πεζούς χαρακτήρες ώστε να είναι άτομα στην Prolog) αν υπάρχει λύση και θα πρέπει να αποτυγχάνει σε περίπτωση που δεν υπάρχει δυνατή διαδρομή.

Παρακάτω δίνονται παραδείγματα πλέγματος, τα αρχεία που τα περιγράφουν και οι εκτελέσεις του προγράμματος σε Java και Prolog.






```
$ cat grid1.txt
5
15 42 3 4 7
17 13 67 4 32
4 10 21 9 6
20 7 4 20 5
105 1 5 3 2
```

```
$ cat grid2.txt
6
88 73 71 65 54 53
90 103 87 76 85 49
4 29 33 38 62 47
101 28 43 67 42 44
27 25 32 47 100 89
6 22 22 17 11 4
```

```
$ cat grid3.txt
2
42 17
17 42
```

Σε Java

```
$ java Moves grid1.txt
[SE,S,SE,SE,E]
$ java Moves grid2.txt
[E,E,E,E,SE,S,SW,NW,W,SW,S,SE,E,E,E]
$ java Moves grid3.txt
IMPOSSIBLE
```

Σε Prolog

```
?- moves("grid1.txt", Moves).
Moves = [se,s,se,se,e].
 ?- moves("grid2.txt", Mvs).
Moves = [e,e,e,e,se,s,sw,nw,w,sw,s,se,e,e,e].
?- moves("grid3.txt", Moves).
false.
```

Δώσε βάση (0.25 βαθμοί)

Εδώ και χρόνια, η ανθρωπότητα είναι συνηθισμένη με το δεκαδικό σύστημα αρίθμησης, το οποίο χρησιμοποιεί δέκα σύμβολα συνολικά (0..9) και για αυτό το λόγο λέμε ότι έχει βάση το δέκα (10). Αλλά φυσικά υπάρχουν και συστήματα αρίθμησης με άλλες βάσεις, όπως το δυαδικό και το δεκαεξαδικό που χρησιμοποιούνται στους υπολογιστές, ή πιο εξωτικά όπως το οκταδικό σύστημα και το τριαδικό σύστημα με βάση το τρία (3) το οποίο, ως πρώτος αριθμός, αποτελεί και τον πιο συνηθισμένο τρόπο αναπαράστασης p-αδικών αριθμών.[1]

Ο γνωστός Βαγγέλης παρατήρησε ότι όταν μετατρέπουμε έναν αριθμό από το δεκαδικό σύστημα σε ένα άλλο μπορούμε πάντα να βρούμε μια βάση τέτοια ώστε ο αριθμός που προκύπτει να γράφεται με ένα μόνο σύμβολο (δηλαδή να έχει όλα τα ψηφία του ίδια). Για παράδειγμα, ο αριθμός 7 έχει το 2 ως τέτοια βάση επειδή η δυαδική του αναπαράσταση είναι 111 ($7_{10} = 111_2$) το 42 έχει το 4 ($42_{10} = 222_4$) ως τέτοια βάση και το 342 το 7 ($342_{10} = 666_7$). Επιπλέον, οι βάσεις αυτές είναι οι ελάχιστες δυνατές.

Η άσκηση σας ζητάει να γράψετε ένα πρόγραμμα σε ML το οποίο να δέχεται ως είσοδο μια (σχετικά μικρού μήκους) λίστα από δεκαδικούς αριθμούς (από το 1 έως το 10^{12}) και να επιστρέφει ως έξοδο μια λίστα ίδιου μήκους με τις *ελάχιστες* βάσεις τέτοιες ώστε η αναπαράσταση του κάθε αριθμού στην αντίστοιχη βάση να έχει μόνο ένα σύμβολο. Δύο παραδείγματα χρήσης της κύριας συνάρτησής σας σε SML/NJ δίνονται παρακάτω:

[1] Άσχετο με την άσκηση, αλλά ενδιαφέρον video: https://youtu.be/tRaq4aYPzCc

```
- minbases [7,42,342];
val it = [2,4,7] : int list
- minbases [123456789];
val it = [11408] : int list
```

Για όσους από εσάς θέλετε να κάνετε την άσκηση σε MLton ή σε OCaml, η είσοδος θα δίνεται από ένα αρχείο το οποίο στην πρώτη γραμμή του θα έχει το πλήθος των αριθμών και στην κάθε μία από τις επόμενες τους αριθμούς. Η έξοδος θα πρέπει να τυπώνει τις αντίστοιχες ελάχιστες βάσεις διαχωρισμένες με newlines (δηλ. σε διαφορετικές γραμμές) όπως φαίνεται παρακάτω:

```
$ cat f1.txt
3
7
42
342
$ cat f2.txt
1
123456789
```

```
$ ./minbases f1.txt
2
4
7
$ ./minbases f2.txt
11408
```

Περαιτέρω οδηγίες για τις ασκήσεις

- 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 σε *όλες τις σειρές ασκήσεων* γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.
- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. 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 ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ ≥ v110.79 ή σε MLton ≥ 20210117 ή σε Objective Caml version ≥ 4.11.1. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.
- Τα προγράμματα σε Java μπορεί να είναι και σε πολλά αρχεία αν θέλετε, αλλά θα πρέπει να περιέχουν μια κύρια κλάση με το όνομα που ζητείται και να δουλεύουν σε Java 17 (17.0.10).
- Τα προγράμματα Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (8.2.4), GNU Prolog (1.4.5) ή YAP (6.2.2), το οποίο θα πρέπει να επιλέξετε.
- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του helios και για να μπορέσετε να τα υποβάλλετε και να βαθμολογηθείτε για αυτά, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχετε εγγραφεί στο μάθημα στο helios. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.