

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 15/7/2024, 23:59:59

Ήταν δίκαιο και έγινε άσκηση, ξανά (0.25 βαθμοί)

Το πρόβλημα με τις δίκαιες υπακολουθίες είναι γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Python και θα πρέπει να έχει την ίδια συμπεριφορά με τα προγράμματα C/C++ και ML στην εκφώνηση εκείνης της σειράς, όπως φαίνεται παρακάτω:

```
$ python3 fairseq.py f1.txt
0
$ python3 fairseq.py f2.txt
1
$ python3 fairseq.py f3.txt
17
```

Δώσε βάση, ξανά (0.25 βαθμοί)

Το πρόβλημα εύρεσης της ελάχιστης βάσης με την οποία μπορεί να αναπαρασταθεί ένας αριθμός με ένα μόνο σύμβολο (ψηφίο του αντίστοιχου συστήματος αρίθμησης) είναι γνωστό από τη δεύτερη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Prolog. Το βασικό κατηγορήμα του προγράμματός σας θα πρέπει να συμπεριφέρεται ως συνάρτηση (δηλ. να δίνει πάντα μία μόνο λύση) όπως φαίνεται παρακάτω:

```
?- minbases([7,42,342], Bases).
Bases = [2,4,7].
?- minbases([123456789], Bases).
Bases = [11408].
```

Επιστροφή στην Αθήνα, ξανά (0.25+0.25 = 0.5 βαθμοί)

Το πρόβλημα της εύρεσης της συντομότερης ακολουθίας κινήσεων για την επιστροφή από το Δάσος του Κόσμου στην Αθήνα σας είναι γνωστό από την δεύτερη σειρά ασκήσεων. Το ζητούμενο εδώ είναι να γραφούν οι λύσεις του σε Python και σε ML. Τα προγράμματά σας θα πρέπει απλά να τυπώνουν τη λύση ως string (ή το **IMPOSSIBLE**) όπως και το πρόγραμμά σας σε Java, όπως φαίνεται παρακάτω:

Σε Python

```
$ python3 moves.py grid1.txt
[SE,S,SE,SE,E]
$ python3 moves.py grid3.txt
IMPOSSIBLE
```

Σε MLton ή σε OCaml

```
$ ./moves grid1.txt
[SE,S,SE,SE,E]
$ ./moves grid3.txt
IMPOSSIBLE
```

Σε SML/NJ

```
- moves "grid1.txt";
[SE,S,SE,SE,E]
val it = () : unit
- moves "grid3.txt";
IMPOSSIBLE
val it = () : unit
```

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 Python πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε Python 3.9.2.
- 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.