

Άσκηση 3

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 10/6/2018, 23:59:59

Ραντεβού στην αγορά, ξανά (0.25 βαθμοί)

Το πρόβλημα με την αγορά της πόλης είναι γνωστό από την πρώτη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του προβλήματος σε Prolog. Επειδή τα συστήματα Prolog δεν τρέχουν native code, ο χρονικός περιορισμός για την άσκηση θα είναι σημαντικά αυξημένος. Το πρόγραμμά σας θα πρέπει να περιέχει ένα κατηγορημα `agora/3` το οποίο θα έχει ως πρώτο όρισμα το όνομα του αρχείου εισόδου και θα επιστρέφει στο δεύτερο και στο τρίτο όρισμά του τη λύση. Για τα παραδείγματα της εκφώνησης της άσκησης, το κατηγορημά σας θα πρέπει να συμπεριφέρεται όπως φαίνεται παρακάτω.¹

```
?- agora('a1.txt', When, Missing) .  
When = 360,  
Missing = 7 ;  
false.  
  
?- agora('a2.txt', When, Missing) .  
When = 840,  
Missing = 0 ;  
false.  
  
?- agora('a3.txt', When, Missing) .  
When = 25845383485350,  
Missing = 4 ;  
false.
```

Για το διάβασμα της εισόδου, δείτε το υπόδειγμα που δίνεται στην τρίτη άσκηση.

Πίστες, κλειδιά και αστέρια, ξανά (0.25 βαθμοί)

Το πρόβλημα με τις πίστες είναι γνωστό από τη δεύτερη σειρά ασκήσεων της φετινής χρονιάς. Το ζητούμενο αυτής της άσκησης είναι να γραφεί η λύση του σε Prolog. (Επειδή τα συστήματα Prolog δεν τρέχουν native code, ο χρονικός περιορισμός για την άσκηση θα είναι σημαντικά αυξημένος.) Το πρόγραμμά σας θα πρέπει να περιέχει ένα κατηγορημα `pistes/2` το οποίο θα έχει ως πρώτο όρισμα το όνομα του αρχείου εισόδου και θα επιστρέφει στο δεύτερό του όρισμα τη λύση. Για τα παραδείγματα της εκφώνησης της άσκησης, το κατηγορημά σας θα πρέπει να συμπεριφέρεται όπως φαίνεται παρακάτω.

```
?- pistes('p1.txt', Answer) .  
Answer = 610 ;  
false.
```

¹ Σε όλα τα παραδείγματα αυτής της σειράς ασκήσεων, ανάλογα με το σύστημα Prolog που θα χρησιμοποιήσετε, στις περιπτώσεις που υπάρχει κάποια λύση, η γραμμή με το `false` μπορεί να λέει `fail` ή `no` ή μπορεί να μην τυπώνεται (που μάλλον είναι το καλύτερο διότι δείχνει ότι η εκτέλεση του κατηγορημάτός σας είναι ντετερμινιστική).

```
?- pistes('p2.txt', Answer).
Answer = 440 ;
false.
```

Για το διάβασμα της εισόδου, δείτε το υπόδειγμα που δίνεται στην τρίτη άσκηση.

Πάμε μουντιάλ... (0.25+0.25 = 0.5 βαθμοί)

Φέτος έχει και πάλι μουντιάλ (το παγκόσμιο πρωτάθλημα ποδοσφαίρου) και οι απανταχού ποδοσφαιρόφιλοι ανασκουμπώνονται. Όπως και στα περισσότερα τουρνουά ποδοσφαίρου, έτσι και στο μουντιάλ η πρώτη φάση γίνεται χωρίζοντας τις ομάδες που συμμετέχουν σε ομίλους και οι επόμενες φάσεις γίνονται με αγώνες νοκ-άουτ. Η άσκηση αυτή ασχολείται μόνο με τους αγώνες νοκ-άουτ. Υποθέτουμε ότι ξεκινάμε με $N=2^k$ ομάδες. Στον πρώτο γύρο των νοκ-άουτ, κάθε ομάδα παίζει με μία άλλη και οι νικητές περνούν στον επόμενο γύρο. Αυτό συνεχίζεται μέχρι τον k -οστό γύρο, όπου έχουν μείνει μόνο οι δύο ομάδες που παίζουν στον τελικό. Υποθέτουμε ότι όλοι οι αγώνες λήγουν υπέρ της μίας ή της άλλης ομάδας, δηλαδή δεν υπάρχουν ισοπαλίες. Στο τέλος του πρωταθλήματος, μας δίνεται ένας πίνακας των ομάδων στον οποίο αναγράφονται οι αγώνες που κάθε ομάδα έπαιξε, τα τέρματα που πέτυχε και αυτά που δέχθηκε, σαν αυτόν που ακολουθεί:

Ομάδα	Αγώνες	Τέρματα υπέρ	Τέρματα κατά
monaco	2	10	2
andorra	2	6	4
sanmarino	1	1	4
liechtenstein	1	0	7

Ο παραπάνω πίνακας μπορεί να αντιστοιχεί στα παρακάτω παιχνίδια:

monaco-andorra	3-2
monaco-liechtenstein	7-0
andorra-sanmarino	4-1

Το πρόγραμμά σας πρέπει να κάνει ακριβώς αυτό: με βάση τον τελικό πίνακα των N ομάδων, πρέπει να βρείτε ένα σύνολο από αποτελέσματα αγώνων που να αντιστοιχούν στον πίνακα. Είναι προφανές ότι η λύση μπορεί να μην είναι μοναδική. Στην περίπτωση αυτή, επιλέξτε όποια λύση θέλετε.

Αυτό που ζητάει λοιπόν η άσκηση είναι να γραφούν δύο προγράμματα (ένα σε Prolog και ένα σε Python 3) τα οποία να παίρνουν ως είσοδο τον τελικό πίνακα και να βρίσκουν ένα δυνατό σύνολο αποτελεσμάτων αγώνων που να είναι συνεπή με αυτόν.

Τα στοιχεία εισόδου θα διαβάζονται από ένα αρχείο με μορφή σαν και αυτή που φαίνεται στα παραδείγματα παρακάτω. Η πρώτη γραμμή του αρχείου έχει έναν ακέραιο αριθμό N , το πλήθος των ομάδων. Οι επόμενες N γραμμές περιέχουν τον πίνακα των ομάδων (μη θεωρήσετε δεδομένη κάποια ταξινόμηση αυτού). Κάθε τέτοια γραμμή περιέχει το όνομα μίας ομάδας t_i (που αποτελείται από το πολύ 20 πεζά γράμματα του λατινικού αλφαβήτου), ακολουθούμενο από τρεις ακέραιους αριθμούς p_i , a_i και b_i — το πλήθος των αγώνων που έπαιξε η ομάδα, το πλήθος των τερμάτων που πέτυχε και το πλήθος των τερμάτων που δέχθηκε, αντίστοιχα.

Το πρόγραμμά σας σε Prolog θα πρέπει να περιέχει ένα κατηγορημα `mundial/2` το οποίο θα παίρνει ως πρώτο όρισμα το όνομα του αρχείου εισόδου και θα επιστρέφει στο δεύτερο όρισμά του τη λύση, δηλαδή τη λίστα των αγώνων στη μορφή που δίνεται στα παραδείγματα παρακάτω. Προσέξτε ότι δεν είναι απαραίτητο να βρίσκει μόνο μία λύση, όμως αν υπάρχουν περισσότερες μόνο η πρώτη λύση σας θα ελεγχθεί και θα βαθμολογηθεί.

Το πρόγραμμά σας σε Python 3 θα πρέπει να διαβάζει την είσοδο από το αρχείο, το όνομα του οποίου δίνεται στη γραμμή εντολών, και να εκτυπώνει τη λίστα των αγώνων στη μορφή που

φαίνεται στα παρακάτω παραδείγματα.

Και για τα δύο προγράμματα, η σειρά των αγώνων στη λίστα εξόδου δεν έχει σημασία, όπως επίσης και η σειρά με την οποία αναφέρονται οι ομάδες στους αγώνες (ποια ομάδα γράφεται πρώτη και ποια δεύτερη). Θεωρήστε δεδομένο ότι το N θα είναι δύναμη του δύο. Η λίστα των αγώνων εξόδου πρέπει να αποτελείται από ακριβώς N-1 αγώνες.

Περιορισμοί: $1 \leq N \leq 64$. Όριο μνήμης: 4GB, όριο χρόνου εκτέλεσης: 1 λεπτό.

Παρακάτω δίνονται κάποια παραδείγματα σε Prolog και σε Python 3.

Σε Prolog

```
?- mundial('m1.txt', Matches).
Matches = [match(monaco, andorra, 3, 2),
            match(monaco, liechtenstein, 7, 0),
            match(andorra, sanmarino, 4, 1)].
```

Σε Python 3

```
$ python3 mundial.py m1.txt
monaco-andorra 3-2
monaco-liechtenstein 7-0
andorra-sanmarino 4-1
```

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

```
$ cat m1.txt
4
monaco 2 10 2
andorra 2 6 4
sanmarino 1 1 4
liechtenstein 1 0 7
```

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με τις προηγούμενες σειρές ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά.
- 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.
- 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 Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (6.6.6), GNU Prolog (1.3.0) ή YAP (6.2.2).
- Τα προγράμματα σε Python πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε Python 3.4.2. (Προσέξτε ότι η Python 2 είναι διαφορετική διάλεκτος της Python!)
- 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 άλλου είδους έξοδο εκτός από τη ζητούμενη διότι δε θα γίνουν δεκτά από το σύστημα υποβολής.