Ημερομηνία Ανάρτησης: 22/11/2016
Ημερομηνία Παράδοσης: 6/12/2016, 23:59μμ
Αρχές Γλωσσών Προγραμματισμού

1. (20%) Ο μικρός Μήτσος θέλει να παίξει Pokemon GO! στο κινητό του. Ένα απο τα χαρακτηριστικά του παιχνιδιού αυτού είναι η λεγόμενη *εξέλιξη (evolution)* των Pokemon. Για να μπορέσει ένα Pokemon είδους P_i να εξελιχθεί, ο Μήτσος θα πρέπει να του δώσει K_i candies τα οποία προορίζονται για το συγκεκριμένο είδος Pokemon. Όταν το Pokemon εξελιχθεί, επιστρέφονται στο Μήτσο 2 candies. Ο Μήτσος έχει N είδη Pokemon και M_i candies για το Pokemon είδους P_i και θέλει να υπολογίσει πόσα Pokemon μπορεί να εξελίξει. Θέλει επίσης να υπολογίσει ποιο Pokemon μπορεί να εξελίξει τις περισσότερες φορές.

   Ορίστε σε Prolog τη σχέση `pokemon(L,N,S)` η οποία παίρνει ως είσοδο μια λίστα `L` και επιστρέφει ως έξοδο τον αριθμό `N` και το όνομα `S`. Πιο συγκεκριμένα, η `L` περιέχει τριάδες που το πρώτο τους στοιχείο είναι το όνομα ενός είδους Pokemon, το δεύτερο στοιχείο είναι ο αριθμός των candies που απαιτούνται για την εξέλιξη του συγκεκριμένου είδους, και το τρίτο στοιχείο είναι ο συνολικός αριθμός από candies που έχει ο Μήτσος για τα Pokemon αυτού του είδους. Ο αριθμός N είναι ο συνολικός αριθμός από Pokemon που ο Μήτσος μπορεί να εξελίξει. Τέλος, `S` είναι το είδος του Pokemon που μπορεί να εξελιχθεί τις περισσότερες φορές (αν υπάρχουν πολλά τέτοια `Pokemon` θα πρέπει να επιστρέφετε αυτό που εμφανίζεται πρώτο σε σειρά στην L). Για παράδειγμα, `pokemon([(caterpie,12,33),(weedle,12,42),(pidgey,12,47),(rattata,25,71)],N,S)` επιστρέφει `N=14, S=weedle`.

2. (20%) Η αεροπορική εταιρία Cantor χρησιμοποιεί το νέο αεροπλάνο της Omega το οποίο έχει ένα άπειρο αριθμό από σειρές καθισμάτων, αριθμημένων με διαδοχικούς φυσικούς αριθμούς ξεκινώντας από το 1. Σε κάθε σειρά υπάρχουν έξι καθίσματα τα οποία είναι αριθμημένα ως εξής: τα καθίσματα 'a', 'b', 'c' βρίσκονται στα αριστερά του διαδρόμου (όπως κοιτάζουμε προς το πιλοτήριο) με το 'a' να βρίσκεται δίπλα στο παράθυρο. Τα καθίσματα 'd', 'e', 'f' βρίσκονται στα δεξιά του διαδρόμου με το 'f' να βρίσκεται δίπλα στο παράθυρο.

   Είναι η ώρα του φαγητού και οι δύο αεροσυνοδοί αρχίζουν να κινούνται από το πιλοτήριο προς την ουρά του αεροπλάνου διατηρώντας μια απόσταση δύο σειρών η μία από την άλλη εξαιτίας του συρόμενου ντουλαπιού που περιέχει τα φαγητά. Έτσι, αρχικά, η πρώτη αεροσυνοδός εξυπηρετεί τη γραμμή 1 ενώ η δεύτερη τη γραμμή 3. Όταν σερβιριστούν όλοι οι επιβάτες και στις δύο σειρές, μετακινούν το ντουλαπάκι κατά μία σειρά: τώρα η πρώτη αεροσυνοδός σερβίρει τη σειρά 2 ενώ η δεύτερη τη σειρά 4. Μετά, μετακινούνται τρεις γραμμές μπροστά και η πρώτη αεροσυνοδός σερβίρει τη σειρά 5 ενώ η δεύτερη τη σειρά 7, κοκ.

   Οι αεροσυνοδοί δουλεύουν με την ίδια ταχύτητα: τους παίρνει 1 δευτερόλεπτο για να σερβίρουν ένα επιβάτη (η Cantor διαλέγει τις αποτελεσματικότερες αεροσυνοδούς για το στόλο της!) και 1 δευτερόλεπτο για να μετακινηθούν κατά μία σειρά. Κάθε αεροσυνοδός πρώτα σερβίρει τους επιβάτες που κάθονται στα δεξιά του διαδρόμου (όπως κοιτάζουμε προς το πιλοτήριο) και μετά αυτούς που κάθονται αριστερά. Επιπλέον, οι επιβάτες σερβίρονται ξεκινώντας από το παράθυρο και πηγαίνοντας προς το διάδρομο.

   Ο μικρός Μήτσος πεινάει και θέλει να υπολογίσει σε πόση ώρα θα πάρει το φαγητό του. Ορίστε σε Prolog τη σχέση `mylunch(R,S,T)` η οποία παίρνει ως είσοδο τον αριθμό γραμμής `R` και τη θέση `S` στην οποία βρίσκεται ο Μήτσος, και επιστρέφει ως έξοδο τον αριθμό των δευτερολέπτων `T` που θα πρέπει να περιμένει ο μικρός μας φίλος πριν αρπάξει το δίσκο από τα χέρια της αεροσυνοδού. Για παράδειγμα, `mylunch(1,f,T)` θα επιστρέψει T=1, `mylunch(2,d,T)` θα επιστρέψει T=10, `mylunch(4,a,T)` θα επιστρέψει T=11, και `mylunch(5,e,T)` θα επιστρέψει T=18.

3. (20%) Έστω `L` μια λίστα με N ακέραιους αριθμούς. Μπορούμε να τροποποιήσουμε τη λίστα με τον ακόλουθο τρόπο: διαλέγουμε δύο γειτονικά στοιχεία της λίστας και τα αντικαθιστούμε με το άθροισμά τους. Με τον τρόπο αυτό τα στοιχεία της λίστας μειώνονται κατά ένα. Θέλουμε να ξέρουμε ποιος είναι ο ελάχιστος αριθμός τροποποιήσεων που πρέπει να κάνουμε ώστε η λίστα να γίνει παλινδρομική. Ορίστε σε Prolog τη σχέση `palindromic(L,N)` η οποία δεδομένης μιας λίστας `L` από ακεραίους, επιστρέφει τον ελάχιστο αριθμό `N` από τροποποιήσεις που μπορούν να κάνουν τη λίστα παλινδρομική. Για παράδειγμα, `palindromic([1,4,3,2],N)` θα επιστρέψει `N=2` διότι `[1,4,3,2]`→`[5,3,2]`→`[5,5]`.

4. (20%) Έστω ότι θέλουμε να κατασκευάσουμε ένα $n \times n$, $n > 2$, πίνακα ο οποίος περιέχει θετικούς ακέραιους και ικανοποιεί τις παρακάτω ιδιότητες:

   - Ο μέσος όρος των αριθμών σε κάθε γραμμή είναι ένας αριθμός που βρίσκεται στη γραμμή αυτή.

- Ο μέσος όρος των αριθμών σε κάθε στήλη είναι ένας αριθμός που βρίσκεται στη στήλη αυτή.
- Όλοι οι αριθμοί στον πίνακα είναι διαφορετικοί.

Ορίστε σε Prolog τη σχέση `mymatrix(N,M)` η οποία παίρνει ως είσοδο έναν ακέραιο $N > 2$ και επιστρέφει ένα πίνακα που έχει τις παραπάνω ιδιότητες. Οποιοσδήποτε πίνακας που ικανοποιεί τις προδιαγραφές είναι αποδεκτός. Για παράδειγμα, `mymatrix(3,M)` θα επιστρέψει `M=[[1,2,3],[4,5,6],[7,8,9]]` (κάθε εσωτερική λίστα αναπαριστά μια γραμμή του πίνακα). Παρατήρηση: στο 20% των test cases με τα οποία θα ελεγχθεί το πρόγραμμά σας, το N θα είναι περιττός αριθμός και στο 80% άρτιος.

5. Ο μικρός Μήτσος παίρνει το πρωί το τραίνο για να πάει στο σχολείο του. Το τραίνο είναι λίγο περίεργο: αποτελείται από μία και μοναδική γραμμή από καθίσματα που το ένα είναι πίσω από το άλλο. Ένα *καλό κάθισμα* ορίζεται ως ένα κάθισμα που έχει, είτε ακριβώς μπροστά του είτε ακριβώς πίσω του (ή και τα δύο), μία κενή θέση. Κάθε 1 λεπτό, ένας νέος επιβάτης μπαίνει στο τραίνο, μέχρι να γεμίσουν όλες οι θέσεις. Τόσο ο Μήτσος όσο και οι υπόλοιποι επιβάτες που μπαίνουν μετά από αυτόν, διαλέγουν τη θέση τους με βάση τους ακόλουθους κανόνες:

   > *Οι καλύτερες θέσεις είναι η πρώτη και η τελευταία, γιατί αυτές θα μείνουν για πάντα καλές (αφού στη μία δεν υπάρχει άλλο κάθισμα μπροστά της και στην άλλη δεν υπάρχει κάθισμα πίσω της). Αν και οι δύο αυτές θέσεις είναι διαθέσιμες, ο επιβάτης που μπαίνει θα διαλέξει την δεξιότερη (δηλαδή την τελευταία). Αν καμία από τις δύο αυτές θέσεις δεν είναι διαθέσιμη, τότε ο επιβάτης που μπαίνει θα διαλέξει εκείνη τη θέση η οποία θα παραμείνει 'καλή' για όσο το δυνατόν περισσότερο χρόνο. Αν πολλές θέσεις θα παραμείνουν καλές για τον ίδιο μέγιστο χρόνο, τότε διαλέγει την δεξιότερη από αυτές.*

   Ο Μήτσος θέλει να επιλέξει τη θέση του γνωρίζοντας ότι και οι επόμενοι επιβάτες που θα μπουν στο τραίνο θα ακολουθήσουν την ίδια ακριβώς προσέγγιση για την επιλογή θέσης όπως και αυτός. Ορίστε σε Prolog τη σχέση `myseat(L,M)` η οποία παίρνει ως είσοδο μια λίστα `L` η οποία περιέχει τη διάταξη των καθισμάτων τη στιγμή που ο Μήτσος μπαίνει στο τραίνο, και επιστρέφει ως έξοδο τη θέση `M` στην οποία θα καθίσει ο μικρός μας φίλος (η αρίθμηση των θέσεων ξεκινάει από το 0). Η λίστα `L` αποτελείται από μια ακολουθία από σταθερές οι οποίες είναι είτε η σταθερά `e` που δείχνει ότι μια θέση είναι άδεια, είτε η σταθερά `o` που δείχνει ότι η αντίστοιχη θέση είναι κατειλημμένη. Για παράδειγμα, η ερώτηση `myseat([e,e,o,e,e], M)` θα επιστρέψει `M=4` (αφού και η πρώτη αλλά και η τελευταία θέση είναι διαθέσιμες, θα διαλέξει την τελευταία). Η ερώτηση `myseat([o,e,e,e,e,o], M)` θα επιστρέψει `M=2`. Η δικαιολόγηση για το τελευταίο παράδειγμα έχει ως εξής. Ο Μήτσος διαλέγει εκείνο το κάθισμα το οποίο θα παραμείνει 'καλό' για τρία λεπτά. Πιο συγκεκριμένα, όταν ο Μήτσος καθίσει στη θέση 2, η διάταξη στο τραίνο γίνεται `[o,e,m,e,e,o]` όπου `m` είναι η θέση που έκατσε ο μικρός ήρωας. Μετά από ένα λεπτό, ένας νέος επιβάτης φτάνει και η διάταξη γίνεται `[o,e,m,e,o,o]`. Μετά από δύο λεπτά η διάταξη θα γίνει `[o,e,m,o,o,o]` και μετά από τρία λεπτά η διάταξη καταλήγει να είναι `[o,o,m,o,o,o]`. Μπορείτε εύκολα να διαπιστώσετε ότι η θέση 2 ήταν η βέλτιστη επιλογή για το Μήτσο.

Παράδοση Ασκήσεων: Η παράδοση πρέπει να γίνει μέχρι τις 23:59μμ, την 6/12/2016. Θα δημιουργήσετε **ένα** αρχείο το οποίο θα περιέχει τις λύσεις όλων των ασκήσεων το οποίο θα στείλετε με *email* και στις τρεις παρακάτω διευθύνσεις: `antru@di.uoa.gr`, `gspapajim@di.uoa.gr`, και `prondo@di.uoa.gr`. Ερωτήσεις σχετικά με τις ασκήσεις θα πρέπει να απευθύνονται στα πρώτα δύο mail (Αντώνης Τρουμπούκης και Γιώργος Παπαδημητρίου). Τα ονόματα των κατηγορημάτων που θα χρησιμοποιήσετε στα προγράμματά σας πρέπει να είναι **ακριβώς** τα ίδια με αυτά που καθορίζονται από την παραπάνω εκφώνηση. Καθυστερημένες ασκήσεις δεν θα βαθμολογηθούν.

Σημείωση: Για να μπορέσει κάποιος να λάβει μέρος στην τελική εξέταση του μαθήματος, θα πρέπει να έχει παραδώσει τις δύο πρώτες εργασίες (Prolog και Haskell) με προβιβάσιμο βαθμό.