Ασκήσεις Φυλλάδιο 9

ECLiPSe CLP: Ο περιορισμός element/3 και Αναζήτηση branch_and_bound.

1. (παράδειγμα διαφανειών) Σε ένα έργο πληροφορικής υπάρχουν 10 εργασίες και 10 ομάδες. Κάθε ομάδα είναι ικανή να εκτελέσει ένα υποσύνολο από τις διαθέσιμες εργασίες. Σε κάθε ομάδα πρέπει να ανατεθεί μια εργασία. Οι ομάδες εκτελούν τις εργασίες με διαφορετικό κόστος. Για παράδειγμα, αν η ομάδα 1 εκτελέσει την εργασία 5 τότε το κόστος είναι 20, ενώ αν η ομάδα 2 εκτελέσει το εργασία 5 το κόστος είναι 70. Οι πληροφορίες για τις ομάδες δίνονται από τα ακόλουθα γεγονότα, όπου το πρώτο όρισμα είναι ο αριθμός της ομάδας, το δεύτερο η εργασίες που μπορεί να κάνει και το τρίτο το αντίστοιχο κόστος ανά εργασία:

```
worker(1,[4,1,3,5,6],[30,8,30,20,10]).
worker(2,[6,3,5,2,4],[140,20,70,10,90]).
worker(3,[8,4,5,7,10],[60,80,10,20,30]).
worker(4,[3,7,8,9,1],[30,40,10,70,10]).
worker(5,[7,1,5,6,4],[40,10,30,20,10]).
worker(6,[8,4,7,9,5],[20,100,130,220,50]).
worker(7,[5,6,7,4,10],[30,30,30,20,10]).
worker(8,[2,6,10,8,3],[50,40,20,10,60]).
worker(9,[1,3,10,9,6],[50,40,10,20,30]).
worker(10,[1,2,7,9,3],[20,20,30,40,50]).
```

Ποια είναι η ανάθεση εργασιών στις ομάδες ώστε το κόστος να είναι ελάχιστο; Να χρησιμοποιήσετε τα γεγονότα, ώστε ο κώδικας να είναι (α) γενικός - αλλάζοντας τα γεγονότα βρίσκει λύση για το νέο πρόβλημα, (β) να λειτουργεί με οσεσδήποτε ομάδες και εργασίες.

2. Έστω 2 αριθμοί πέντε ψηφίων ο καθένας, οι οποίοι είναι μερικώς γνωστοί, δηλαδή ξέρω μερικά από τα ψηφία και τις θέσεις τους, όπως φαίνεται παρακάτω:

[?,5,?,?,3] [?,?,0,?,1]

Γνωρίζω ότι κάθε ψηφίο από 0.. 9 εμφανίζεται μόνο μια φορά στους δύο αριθμούς. Ποια είναι τα υπόλοιπα ψηφία, ώστε η διαφορά των αριθμών ***να είναι η ελάχιστη δυνατή***? Το κατηγόρημα **num_gen_min/3** που θα δημιουργήσετε θα έχει ένα ακόμη όρισμα το οποίο θα είναι η διαφορά των δύο αριθμών. Για παράδειγμα:

```
?- num_gen_min(N1, N2, D).
N1 = [6, 5, 9, 8, 3]
N2 = [7, 2, 0, 4, 1]
D = 6058
Yes
```

3. Σε ένα πανεπιστήμιο θα πρέπει να ανατεθούν 10 διπλωματικές εργασίες σε αντίστοιχους 10 φοιτητές. Ο κάθε φοιτητής μπορεί να δηλώσει μέχρι 5 εργασίες με σειρά προτίμησης. Για παράδειγμα ο φοιτητής 1, δήλωσε ότι επιθυμεί να αναλάβει τις εργασίες **4,1,3,5,6,** με σειρά προτίμησης αυτή που εμφανίζεται. Οι επιλογές που έκαναν οι φοιτητές φαίνονται παρακάτω

```
student(alex,[4,1,3,5,6]).
student(nick,[6,3,5,2,4]).
student(jack,[8,4,5,7,10]).
student(helen,[3,7,8,9,1]).
student(maria,[7,1,5,6,4]).
student(evita,[8,4,7,9,5]).
student(jacky,[5,6,7,4,10]).
student(peter,[2,6,10,8,3]).
student(john,[1,3,10,9,6]).
student(mary,[1,6,7,9,10]).
```

Να αναπτύξετε ένα Prolog πρόγραμμα (με top-level κατηγόρημα το **solve(S,Cost)**) το οποίο κάνει την καλύτερη ανάθεση διπλωματικών εργασιών στους φοιτητές δεδομένων των επιλογών τους. Όλοι οι φοιτητές είναι ίσοι, και οι επιλογές που γίνονται θα πρέπει να δίνουν (όσο είναι δυνατό) την εργασία που προτιμά ο φοιτητής περισσότερο. Παράδειγμα εκτέλεσης είναι το ακόλουθο:

```
?- solve_thesis(Students, C).
Students = [(alex, 4), (nick, 6), (jack, 8), (helen, 3), (maria,
7), (evita, 9), (jacky, 5), (peter, 2), (john, 10), (mary, 1)]
C = 15
Yes (0.01s cpu)
```

4. Σε μια εταιρεία, πρέπει να μεταφερθούν κάποια από τα 10 κιβώτια διαφορετικού βάρους ιδίου όγκου, από δύο φορτηγά. Το φορτηγό Α μπορεί να μεταφέρει 3 κιβώτια με μέγιστο βάρος 1200kg, ενώ το δεύτερο 4 κιβώτια με μέγιστο βάρος 1350kg. Τα βάρη των κιβωτίων δίνονται από γεγονότα της μορφής:

```
box(1,140).
box(2,200).
box(3,450).
box(4,700).
box(5,120).
box(6,300).
box(7,250).
box(8,125).
box(9,600).
```

```
box(10,650).
```

Να αναπτύξετε ένα CLP πρόγραμμα το θα έχει ένα κατηγόρημα **load_trucks/4** το οποίο να βρίσκει ποια κιβώτια θα πρέπει να μεταφέρει κάθε φορτηγό και το συνολικό βάρος των κιβωτίων, για να εκμεταλλευόμαστε όσο το δυνατό περισσότερο το ωφέλιμο φορτίο τους. Για παράδειγμα, στην ακόλουθη εκτέλεση, TA είναι τα κιβώτια που θα μεταφέρει το φορτηγό A, και LA το βάρος τους, ενώ TB και LB τα αντίστοιχα μεγέθη για το φορτηγό B:

```
?- load_trucks(TA,LA,TB,LB).
TA = [2, 4, 6]
LA = 1200
TB = [3, 5, 8, 10]
LB = 1345
Yes
```

5. Τρεις πάροχοι προσφέρουν διαδικτυακό χώροι αποθήκευσης (cloud storage) σε διαφορετικές χωρητικότητες (GB) και σε διαφορετικές τιμές, που εκφράζεται με τα ακόλουθα γεγονότα:

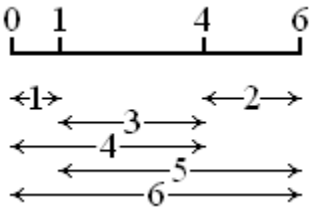
```
provider(a,[0,750,1000,1500],[0,10,13,17]).
provider(b,[0,500,1250,2000],[0,8,12,22]).
provider(c,[0,1000,1750,2000],[0,15,18,25]).
provider(d,[0,1000,1500,1750],[0,13,15,17]).
```

όπου το πρώτο όρισμα είναι το όνομα της εταιρείας, το δεύτερο οι επιλογές του αποθηκευτικού χώρου και το τρίτο οι αντίστοιχες τιμές. Για παράδειγμα η εταιρεία a, προσφέρει 750 GB, με 10 ευρώ. Έστω ότι (α) ο συνολικός χώρος που θα πρέπει να αποκτήσουμε είναι μεγαλύτερος από 3600 GB και μικρότερος από 4600 GB, (β) η συνολική τιμή θα πρέπει να είναι η μικρότερη δυνατή, και (γ) μπορεί να γίνει μόνο ένα συμβόλαιο με κάθε εταιρεία. Να υλοποιήσετε ένα κατηγόρημα **space/2** το οποίο βρίσκει με ποιον provider και ποιο συμβόλαιο πρέπει να επιλέξουμε για να ικανοποιήσουμε τις παραπάνω απαιτήσεις.

Η λύση επιστρέφεται σαν λίστα, με το συμβόλαιο που πρέπει να γίνει με τους providers a,b,c,d αντίστοιχα:

```
?- space(I, C).
I = [0, 2000, 0, 1750]
C = 39
Yes (0.00s cpu)
```

6. Μια σειρά από κεραίες πρέπει να τοποθετηθούν στη σειρά έτσι ώστε η απόσταση μεταξύ δύο κεραιών να είναι διαφορετική από την απόσταση μεταξύ δύο οποιονδήποτε άλλων κεραιών. Αν μετράμε η πρώτη κεραία είναι στη θέση 0, τότε ποια είναι **η ελάχιστη απόσταση** που μπορεί να τοποθετηθεί η τελευταία κεραία? Για παράδειγμα το ακόλουθο σχήμα δείχνει την τοποθέτηση 4 κεραιών σύμφωνα με τον παραπάνω κανόνα:



Να υλοποιήσετε ένα πρόγραμμα το οποίο βρίσκει τις θέσεις των κεραιών για διαφορετικούς αριθμούς κεραιών και επιστρέφει την ελάχιστη απόσταση. Για παράδειγμα

```
?- antennas(4, Max, L).
Max = 6
L = [0, 1, 4, 6]
Yes

?- antennas(5, Max, L).
Max = 11
L = [0, 1, 4, 9, 11]
Yes

?- antennas(7, Max, L).
Max = 25
L = [0, 1, 4, 10, 18, 23, 25]
Yes

?- antennas(8, Max, L).
Max = 34
L = [0, 1, 4, 9, 15, 22, 32, 34]
Yes
```

7. Θα πρέπει να ενοικιάσετε υπολογιστικούς πόρους στο νέφος (cloud) ώστε να τρέξετε μια μεγάλη σειρά πειραμάτων μηχανικής μάθησης. Για να τρέξετε τα πειράματά σας, θα πρέπει να έχετε *δύο* εικονικές μηχανές (virtual machines), *με τουλάχιστον 4GB μνήμης RAM* η κάθε μια και θα πρέπει να υπάρχουν *συνολικά στις δύο εικονικές μηχανές ακριβώς 12 vcpus* (Virtual CPUs) για να μπορείτε να μοιράσετε τις εργασίες. Προφανώς θέλετε την ελάχιστη δυνατή συνολική τιμή. Οι τρέχουσες

προσφορές από τους παρόχους δίνονται στα ακόλουθα γεγονότα:

```
ram([2,8,8,16,2,4]).
price([30,35,20,38,44,65]).
vcpu([4,8,8,4,4,8]).
```

Δηλαδή, η πρώτη εικονική μηχανή έχει 2GB Ram, 4 vcpus και κοστίζει 30 ευρώ. Να γράψετε ένα CLP πρόγραμμα **select_providers(X,Y,Price)** το οποίο επιστρέφει τις δύο προσφορές που επιλέξατε και την συνολική τους τιμή. Παράδειγμα εκτέλεσης του προγράμματος δίνεται παρακάτω:

```
?- select_providers(X, Y, Price).
X = 3
Y = 4
Price = 58
Yes (0.00s cpu)
```