

Λογικός Προγραμματισμός με Περιορισμούς

Τμ. Εφαρμοσμένης Πληροφορικής

Εργασία 2 2020-2021

(υπάρχουν συνολικά 3 ασκήσεις)

1. Cloud Virtual Machines (35/100)

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 διαφορετικές 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, έτσι ώστε αυτή να είναι η ελάχιστη δυνατή. Παράδειγμα εκτέλεσης του προγράμματος δίνεται παρακάτω:

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

Σημ. Ο κώδικας σας πρέπει να είναι γενικός, δηλαδή να λειτουργεί με οποιαδήποτε αριθμό γεγονότων της παραπάνω μορφής.

2. Zoom Classes (40/100)

Ένα τμήμα έχει 100 άδειες zoom τις οποίες μοιράζονται τα μαθήματα. Τα μαθήματα τα οποία διεξάγονται σε μια ημέρα έχουν διαφορετικές απαιτήσεις σε αριθμό αδειών, διαφορετική διάρκεια, και διαφορετικές απαιτήσεις σε διδάσκοντες, όπως δίνεται στα παρακάτω γεγονότα. Στα ακόλουθα το πρώτο όρισμα είναι το όνομα του μαθήματος, το δεύτερο οι ώρες διδασκαλίας του μαθήματος, το τρίτο ο αριθμός των αδειών που απαιτούνται και το τέταρτο οι απαιτήσεις σε διδάσκοντες.

```
class(clp,3,40,3).  
class(procedural,3,60,2).
```

```
class (analysis,4,50,2) .  
class (computer_sys,4,40,3) .  
class (algebra,3,40,4) .
```

Οι περιορισμοί είναι:

- Τα μαθήματα ξεκινούν στις **9.00** και μπορούν να ολοκληρωθούν (να τελειώσουν) στις **21.00**.
- Οι συνολικές άδειες zoom είναι **100**.
- Οι διδάσκοντες είναι **6**. Μπορούν να κάνουν οποιοδήποτε μάθημα.
- Τα μαθήματα θα πρέπει να ολοκληρωθούν στο συντομότερο δυνατό χρόνο.

Να υλοποιήσετε το κατηγορήμα lectures/3 το οποίο βρίσκει το πότε θα ξεκινήσει κάθε μάθημα μέσα στην ημέρα. Παράδειγμα εκτέλεσης:

```
?- lectures (Classes, Starts, Makespan) .  
Classes = [clp, procedural, analysis, computer_sys, algebra]  
Starts = [9, 9, 12, 12, 16]  
Makespan = 19  
Yes
```

Στο παραπάνω, τα μαθήματα clp και procedural ξεκινούν στις 9, κοκ.

Το παραπάνω ΔΕΝ είναι η βέλτιστη λύση.

Σημ. Ο κώδικας σας πρέπει να είναι **γενικός**, δηλαδή να λειτουργεί με οποιαδήποτε αριθμό γεγονότων της παραπάνω μορφής.

3. Manufacturing (25/100)

Μια εταιρεία έχει να παραδώσει 4 διαφορετικές παραγγελίες με διαφορετικό αριθμό προϊόντων, όπως φαίνεται στα ακόλουθα γεγονότα.

```
product (a,40) .  
product (b,56) .  
product (c,48) .  
product (d,64) .
```

Στα προϊόντα θα πρέπει να μπει ετικέτα και αν γίνει ποιοτικός έλεγχος.

Αν:

- υπάρχουν 8 μηχανές που εκτυπώνουν ετικέτες,
- κάθε μηχανή μπορεί να βάλει ετικέτα σε ένα προϊόν το λεπτό,
- κάθε μηχανή απαιτεί 15 λεπτά για παραμετροποιηθεί ώστε να μπορεί να τυπώσει σωστά την ετικέτα,
- δεν μπορεί για καμιά παραγγελία να χρησιμοποιηθούν λιγότερες από 2 μηχανές εκτύπωσης ταυτόχρονα,
- ο ποιοτικός έλεγχος μπορεί να χειριστεί μέχρι 2 παραγγελίες, ασχέτως από πόσες μηχανές εκτυπώνουν και γίνεται ταυτόχρονα με την εκτύπωση,
- οι μηχανές εκτύπωσης που θα ανατεθούν στην παραγγελία πρέπει να λειτουργούν όλες από την αρχή μέχρι το τέλος της παραγγελίας, δηλαδή τα προϊόντα της παραγγελίας σχηματίζουν

ορθογώνιο.

Τα ερωτήματα που θα απαντήσει ο κώδικάς σας είναι:

- πότε θα ξεκινήσει κάθε παραγγελία την εκτύπωση,
- πόσο θα διαρκέσει η εκτύπωση της παραγγελίας σε λεπτά (όχι ο χρόνος παραμετροποίησης)
- πόσες μηχανές θα χρειαστεί η κάθε παραγγελία
- ποιος είναι ο βέλτιστος συνολικός χρόνος για την εκτύπωση όλων των παραγγελιών (σε λεπτά).

Να υλοποιήσετε ένα CLP πρόγραμμα **company(S,D,Machines,M)**, το οποίο απαντά σε όλα τα παραπάνω ερωτήματα, με το **S** να είναι οι χρόνοι έναρξης της κάθε παραγγελίας, **D** οι χρόνοι εκτύπωσης (χωρίς τον χρόνο παραμετροποίησης) του κάθε παραγγελίας, **Machines** οι μηχανές που απαιτούνται για κάθε παραγγελία και **M** ο συνολικός χρόνος εκτύπωσης. Για παράδειγμα:

```
?- company(S, D, Machines, M).  
S = st([0, 0, 23, 43])  
D = dur([8, 28, 24, 32])  
Machines = machines([5, 2, 2, 2])  
M = 75
```

Στο παράδειγμα εκτέλεσης, η πρώτη παραγγελία θα ξεκινήσει την χρονική στιγμή 0, η εκτύπωση των προϊόντων θα διαρκέσει 8 λεπτά και θα απαιτηθούν 5 μηχανές ($5 \cdot 8 = 40$).

Η παραπάνω λύση ΔΕΝ είναι η βέλτιστη.

Σημ. Ο κώδικας σας πρέπει να είναι γενικός, δηλαδή να λειτουργεί με οποιαδήποτε αριθμό γεγονότων της παραπάνω μορφής.

ΠΑΡΑΔΟΣΗ

Στο ECLASS θα βρείτε το αρχείο **exec2.ecl** με όλο τον κώδικα που χρειάζεστε για τις παραπάνω ασκήσεις.

Θα παραδώσετε εντός της ημερομηνίας που αναφέρεται στο ECLASS **ΕΝΑ ZIP ΑΡΧΕΙΟ** με τα ακόλουθα:

- Ένα αρχείο με το όνομα **exec2.ecl** το οποίο θα περιέχει **όλες τις λύσεις των ασκήσεων**.
- Ένα αρχείο **report.pdf** (σε μορφή pdf) το οποίο θα περιέχει:
 - Στην πρώτη σελίδα το Όνομά σας, τον Αριθμό μητρώου σας και το email σας.
 - Για κάθε μια από τις τρεις ασκήσεις:
 - τον **κώδικα** (ασχέτως αν βρίσκεται και στο αντίστοιχο αρχείο) και σχολιασμό σχετικά με αυτόν.
 - Παραδείγματα εκτέλεσης (για κάθε κατηγορία, όπου έχει νόημα)
 - Bugs και προβλήματα που έχει ο κώδικάς σας.

ΠΡΟΣΟΧΗ: ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΚΑΙ ΤΗ ΣΕΙΡΑ ΤΩΝ ΟΡΙΣΜΑΤΩΝ ΠΟΥ ΔΙΝΕΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ)

Καλή επιτυχία (και *have fun with Prolog!*)