

**ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

Β΄ Ομάδα Ασκήσεων "Λογικού Προγραμματισμού"
Ακαδημαϊκού Έτους 2014-15

Οι ασκήσεις της ομάδας αυτής πρέπει να αντιμετωπισθούν με τη βοήθεια της τεχνολογίας του **προγραμματισμού με περιορισμούς**. Ένα σύστημα λογικού προγραμματισμού που υποστηρίζει την τεχνολογία αυτή είναι η **ECLⁱPS^e** (<http://www.eclipseclp.org>). Μπορείτε να χρησιμοποιήσετε είτε την παλιότερη βιβλιοθήκη **fd** είτε την νεώτερη **ic**. Η βιβλιοθήκη **fd** τεκμηριώνεται στο κεφάλαιο 2 του “Obsolete Libraries Manual” και η **ic** στα κεφάλαια 3 και 4 του “Constraint Library Manual”. Αν χρησιμοποιήσετε την **ic**, θα σας χρειαστεί και η βιβλιοθήκη **branch_and_bound**, ιδιαιτέρως το κατηγορήμα **bb_min/3**, το οποίο τεκμηριώνεται, όπως και όλα τα κατηγορήματα που παρέχει η **ECLⁱPS^e**, στο “Alphabetical Predicate Index”.

Εναλλακτικά συστήματα λογικού προγραμματισμού με περιορισμούς που μπορείτε να χρησιμοποιήσετε για τις ασκήσεις αυτής της ομάδας είναι η **GNU Prolog** (<http://www.gprolog.org>) και η **SWI-Prolog** (<http://www.swi-prolog.org>), αλλά δεν προτείνονται, διότι οι βιβλιοθήκες περιορισμών τους είναι περιορισμένης λειτουργικότητας και όχι ιδιαίτερα αποδοτικές.

Σε κάθε περίπτωση, στα αρχεία που θα παραδώσετε, θα πρέπει να αναφέρεται στην αρχή τους, σαν σχόλιο, για ποιο σύστημα Prolog έχουν υλοποιηθεί τα αντίστοιχα προγράμματα, εάν αυτό είναι διαφορετικό από την **ECLⁱPS^e**.

Άσκηση 4

Μία δισδιάστατη ασπρόμαυρη εικόνα έχει σκαναριστεί οριζόντια, κάθετα, και διαγώνια (κατά τις κατιούσες και τις ανιούσες διαγωνίους της) και έχει κωδικοποιηθεί με βάση τα πλήθη των μαύρων pixels σε κάθε γραμμή, κάθε στήλη, κάθε κατιούσα διαγώνιο και κάθε ανιούσα διαγώνιο. Ορίστε ένα κατηγορήμα **decode/4**, το οποίο να δέχεται σαν ορίσματα κατά σειρά τα πλήθη των μαύρων pixels για όλες τις γραμμές, στήλες, κατιούσες και ανιούσες διαγωνίους της εικόνας, να αποκωδικοποιεί την εικόνα και να την εκτυπώνει.

Μπορείτε να χρησιμοποιήσετε αυτούσιο, αν πιστεύετε ότι σαν είναι χρήσιμο, το κατηγορήμα **diags/3** που ορίσατε στην άσκηση 3, αν στο πηγαίο αρχείο με τον ορισμό του **decode/4** γράψετε στην αρχή το:

```
:- [diags].
```

Κάποια παραδείγματα εκτέλεσης είναι τα εξής:

```
?- decode([1,1,10,11,10,1,1],[3,3,3,3,3,3,3,5,5,3,1],
           [0,0,1,2,3,3,3,4,3,4,3,3,2,4,0,0,0],
           [0,0,1,2,3,3,3,4,3,4,3,3,2,4,0,0,0]).
```

```
. . . . . * . . .
. . . . . * . . .
* * * * * * * * *
* * * * * * * * *
* * * * * * * * *
. . . . . * . . .
. . . . . * . . .
```

```
?- decode([4,4,3,3,4,4],[2,2,2,4,2,4,2,2,2],
           [0,2,1,0,2,1,5,5,1,2,0,1,2,0],
           [0,2,1,0,2,1,5,5,1,2,0,1,2,0]).
```

```
. * * . . * * .
* . . * . * . *
. . . * * * . .
. . . * * * . .
* . . * . * . *
. * * . . * * .
```

```
?- decode([13,10,14,10,13],
           [5,2,2,1,0,5,2,3,2,0,3,2,2,3,0,5,1,1,1,0,3,2,2,3,
            0,3,2,3,2],
           [1,1,1,2,2,2,3,1,4,2,1,4,2,1,1,3,4,2,2,1,0,2,2,1,
            1,2,4,2,1,2,2,1,0],
           [1,2,2,2,3,1,2,2,2,3,1,2,3,1,1,3,3,1,1,1,1,3,3,1,
            1,2,4,2,1,1,2,2,0]).
```

```
* * * . . * * * . . * * . . * . . . . * * . . * * .
* . . * . * . . * . * . . * . . . . * . . * . * . .
* * * . . * * * . . * . . * . . . . * . . * . * *
* . . . . * . * . . * . . * . . . . * . . * . * .
* . . . . * . . * . . * * . . * * * * . . * * . . * * .
```

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο Prolog με όνομα **decode.pl**.

Άσκηση 5

Το πρόβλημα της κατασκευής δυαδικών ακολουθιών χαμηλής αυτοσυσχέτισης (low autocorrelation binary sequences) είναι πολύ σημαντικό για διάφορες περιοχές εφαρμογών από τις τηλεπικοινωνίες, τη φυσική, τη χημεία, κλπ. Το ζητούμενο στο πρόβλημα αυτό είναι να κατασκευασθεί μία ακολουθία από -1 και $+1$, δεδομένου μήκους n , τέτοια ώστε το άθροισμα των τετραγώνων των αυτοσυσχετίσεων k -τάξης ($0 < k < n$) της ακολουθίας να είναι το ελάχιστο δυνατό. Συγκεκριμένα:

Έστω ότι η ζητούμενη ακολουθία είναι η S_i , όπου $0 \leq i < n$ και $S_i \in \{-1, +1\}$. Η αυτοσυσχέτιση k -τάξης C_k της ακολουθίας ορίζεται ως

$$C_k = \sum_{i=0}^{n-k-1} S_i \cdot S_{i+k},$$

όπου $0 < k < n$. Ζητούμε ακολουθία μήκους n , για την οποία η ολική αυτοσυσχέτισή της

$$E = \sum_{k=1}^{n-1} C_k^2$$

να είναι ελάχιστη.

Ορίστε ένα κατηγορημα `labs/3`, το οποίο όταν καλείται σαν `labs(N, Labs, AC)`, να επιστρέφει στη λίστα `Labs` ακολουθία μήκους `N` ελάχιστης αυτοσυσχέτισης. Το κατηγορημα να επιστρέφει και την τιμή `AC` της ελάχιστης αυτοσυσχέτισης. Ενδεικτικές εκτελέσεις είναι οι εξής:

```
?- labs(3, Labs, AC).
Labs = [-1, -1, 1]
AC = 1
```

```
?- labs(4, Labs, AC).
Labs = [-1, -1, -1, 1]
AC = 2
```

```
?- labs(5, Labs, AC).
Labs = [-1, -1, -1, 1, -1]
AC = 2
```

```
?- labs(7, Labs, AC).
Labs = [-1, -1, -1, 1, 1, -1, 1]
AC = 3
```

```
?- labs(10, Labs, AC).
Labs = [-1, -1, -1, -1, -1, 1, 1, -1, 1, -1]
AC = 13
```

```
?- labs(13, Labs, AC).
Labs = [-1, -1, -1, -1, -1, 1, 1, -1, -1, 1, -1, 1, -1]
AC = 6
```

```
?- labs(15, Labs, AC).
Labs = [-1, -1, -1, -1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, 1]
AC = 15
```

```
?- labs(17, Labs, AC).
Labs = [-1, -1, -1, -1, 1, 1, -1, -1, 1, -1, -1, 1, -1, 1, -1,
        1, 1]
AC = 32
```

```
?- labs(18, Labs, AC).
Labs = [-1, -1, -1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, -1, 1,
        1, -1, -1]
AC = 25
```

```
?- labs(19, Labs, AC).
Labs = [-1, -1, -1, -1, 1, -1, 1, -1, 1, 1, 1, 1, -1, -1, 1,
        1, -1, 1, 1]
AC = 29

?- labs(20, Labs, AC).
Labs = [-1, -1, -1, -1, -1, 1, -1, 1, 1, 1, -1, 1, -1, -1, 1,
        1, 1, -1, -1, 1]
AC = 26
```

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο **Prolog** με όνομα **labs.pl**.

Άσκηση 6

Μία εταιρεία επιθυμεί να κατασκευάσει έναν αριθμό αποθηκών, από τις οποίες θα μπορεί να τροφοδοτεί τους πελάτες της με τα προϊόντα που παράγει. Από κάποια αρχική μελέτη που έγινε, προέκυψαν διάφορες υποψήφιες θέσεις κατασκευής αποθηκών, έστω πλήθους N . Το κόστος κατασκευής αποθήκης σε κάποια από τις υποψήφιες θέσεις είναι δεδομένο και ισούται με F_i , για κάθε i από 1 μέχρι N . Έστω ότι η εταιρεία προμηθεύει M πελάτες. Το κόστος εξυπηρέτησης του πελάτη j από την αποθήκη i , εφ' όσον αυτή κατασκευασθεί τελικά, είναι δεδομένο και ισούται με C_{ij} , για κάθε i από 1 μέχρι N και κάθε j από 1 μέχρι M .

Το ζητούμενο του προβλήματος είναι να αποφασισθεί σε ποιες από τις υποψήφιες θέσεις πρέπει η εταιρεία να κατασκευάσει αποθήκες και από ποια αποθήκη θα εξυπηρετείται κάθε πελάτης, έτσι ώστε το συνολικό κόστος κατασκευής των αποθηκών και εξυπηρέτησης των πελατών να είναι το ελάχιστο δυνατό για την εταιρεία.

Δεδομένα για το πρόβλημα, με $N = 20$ υποψήφιες θέσεις για αποθήκες και $M = 40$ πελάτες, βρίσκονται στο http://www.di.uoa.gr/~takis/warehouses_data.pl. Στο αρχείο αυτό, με το κατηγορήμα `fixedcosts/1` δίνονται, στη μορφή λίστας, τα κόστη κατασκευής των αποθηκών στις διάφορες υποψήφιες θέσεις και με το κατηγορήμα `varcosts/1` δίνονται τα κόστη εξυπηρέτησης των πελατών, σαν μία λίστα από λίστες. Κάθε εσωτερική λίστα αντιστοιχεί σε έναν πελάτη, τα στοιχεία της οποίας είναι τα κόστη εξυπηρέτησης του πελάτη από όλες τις υποψήφιες θέσεις αποθηκών.

Γράψτε ένα κατηγορήμα `warehouses/5`, το οποίο να καλείται σαν `warehouses(N1, M1, YesNoLocs, CustSrvs, Cost)`, και λαμβάνοντας υπόψη μόνο τις $N1$ πρώτες υποψήφιες θέσεις και τους $M1$ πρώτους πελάτες από τα δεδομένα του αρχείου που δόθηκε, να βρίσκει **λύση ελαχίστου κόστους** του προβλήματος, επιστρέφοντας στην μεταβλητή `YesNoLocs` μία λίστα (μήκους $N1$) από 1 και 0, όπου το 1 σημαίνει ότι η αντίστοιχη αποθήκη θα κατασκευασθεί, ενώ 0 ότι δεν θα κατασκευασθεί, στην μεταβλητή `CustSrvs` μία λίστα (μήκους $M1$) από τους αριθμούς αποθηκών από τις οποίες εξυπηρετούνται οι πελάτες και στην μεταβλητή `Cost` το συνολικό κόστος της λύσης. Αν κάποιο από τα $N1$ ή/και $M1$ δοθεί ίσο με 0, τότε να ληφθούν υπόψη όλες οι υποψήφιες θέσεις ή/και όλοι οι πελάτες, αντίστοιχα, που δίνονται στο αρχείο. Κάποιες ενδεικτικές εκτελέσεις είναι οι εξής:

```

?- warehouses(5, 10, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [0, 1, 1, 0, 1]
CustSrvs = [5, 5, 5, 3, 5, 5, 2, 3, 5, 5]
Cost = 105452

?- warehouses(5, 15, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [0, 1, 1, 1, 1]
CustSrvs = [4, 4, 5, 3, 4, 5, 2, 3, 4, 4, 4, 5, 5, 5, 5]
Cost = 156604

?- warehouses(10, 15, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [0, 1, 0, 1, 1, 1, 1, 0, 0, 0]
CustSrvs = [7, 7, 5, 6, 4, 5, 2, 6, 4, 4, 4, 5, 6, 5, 7]
Cost = 148880

?- warehouses(15, 20, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
CustSrvs = [8, 7, 5, 6, 8, 5, 2, 6, 8, 8, 4, 5, 6, 5, 7,
            8, 4, 9, 4, 7]
Cost = 192820

?- warehouses(20, 30, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0,
            0, 0, 0, 0]
CustSrvs = [12, 12, 1, 6, 12, 1, 2, 6, 12, 12, 4, 5, 6, 1, 7,
            12, 4, 9, 4, 7, 4, 7, 5, 1, 12, 5, 13, 5, 5, 1]
Cost = 385634

?- warehouses(0, 0, YesNoLocs, CustSrvs, Cost).
YesNoLocs = [1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1,
            1, 1, 0, 0]
CustSrvs = [8, 12, 1, 18, 8, 1, 2, 6, 8, 8, 4, 5, 6, 1, 12,
            8, 4, 9, 4, 12, 4, 16, 5, 1, 12, 5, 13, 5, 5,
            1, 16, 5, 16, 17, 12, 12, 18, 6, 8, 18]
Cost = 643420

```

Παραδοτέο για την άσκηση είναι ένα **πηγαίο αρχείο Prolog** με όνομα **warehouses.pl**.

Άσκηση 7

Έστω ότι έχουμε N άντρες και N γυναίκες και το ζητούμενο είναι να τους παντρέψουμε με τέτοιο τρόπο ώστε να προκύψουν N σταθεροί γάμοι. Κάθε άντρας προτιμά σε διαφορετικό βαθμό τις υποψήφιες συζύγους του και κάθε γυναίκα επίσης έχει διαβαθμίσει τους υποψήφιους συζύγους της. Ένας γάμος A - Γ είναι σταθερός όταν για κάθε γυναίκα $\Gamma 1$ που ο άντρας A προτιμά περισσότερο από τη σύζυγό του Γ αυτή προτιμά το σύζυγό της $A1$ περισσότερο από τον A και για κάθε άντρα $A2$ που η γυναίκα Γ προτιμά περισσότερο από το σύζυγό της A αυτός προτιμά τη σύζυγό του $\Gamma2$ περισσότερο από τη Γ . Αν δουλέψετε με τη βιβλιοθήκη `fd` της ECLⁱPS^e, δεδομένα θα βρείτε στο http://www.di.uoa.gr/~takris/stable_data.pl (για την `ic`, στο http://www.di.uoa.gr/~takris/stableic_data.pl). Ορίστε ένα κατηγορημα

stable/1, το οποίο να επιστρέφει μία λύση του προβλήματος, και μέσω οπισθοδρόμησης όλες τις λύσεις τελικά, σαν μία λίστα από γάμους Α-Γ. Έστω ότι τα δεδομένα (από το http://www.di.uoa.gr/~takís/stable_data.pl) για το πρόβλημα είναι τα εξής:

```
men([rick,jack, john,hugh,greg,nick,bill,andy,alan,dick]).

women([helen,tracy,linda,sally,wanda,maria,diana,patty,jenny,lilly]).

prefers(rick,[tracy,linda,jenny,maria,wanda,sally,diana,patty,helen,lilly]).
prefers(jack,[tracy,lilly,patty,sally,diana,linda,jenny,helen,maria,wanda]).
prefers(john,[diana,jenny,wanda,linda,patty,tracy,sally,helen,lilly,maria]).
prefers(hugh,[helen,wanda,diana,maria,sally,patty,linda,lilly,jenny,tracy]).
prefers(greg,[jenny,maria,lilly,patty,sally,linda,tracy,diana,helen,wanda]).
prefers(nick,[sally,linda,diana,maria,jenny,lilly,wanda,helen,patty,tracy]).
prefers(bill,[linda,tracy,diana,patty,lilly,sally,jenny,helen,wanda,maria]).
prefers(andy,[helen,jenny,lilly,maria,sally,patty,wanda,tracy,diana,linda]).
prefers(alan,[patty,sally,maria,jenny,linda,diana,helen,lilly,tracy,wanda]).
prefers(dick,[sally,wanda,helen,maria,lilly,diana,jenny,tracy,patty,linda]).
prefers(helen,[alan,rick,nick,bill,jack,hugh,dick,john,andy,greg]).
prefers(tracy,[andy,john,alan,hugh,bill,rick,greg,dick,jack,nick]).
prefers(linda,[dick,hugh,john,nick,andy,greg,alan,jack,bill,rick]).
prefers(sally,[nick,dick,bill,rick,greg,andy,jack,hugh,alan,john]).
prefers(wanda,[greg,bill,dick,jack,john,nick,alan,andy,rick,hugh]).
prefers(maria,[john,alan,nick,rick,greg,hugh,jack,bill,andy,dick]).
prefers(diana,[greg,hugh,andy,nick,john,bill,alan,dick,jack,rick]).
prefers(patty,[alan,bill,jack,hugh,greg,dick,rick,nick,andy,john]).
prefers(jenny,[greg,dick,jack,bill,alan,john,andy,hugh,rick,nick]).
prefers(lilly,[hugh,jack,dick,nick,andy,alan,greg,rick,bill,john]).
```

Τότε, μία ενδεικτική εκτέλεση του ζητούμενου κατηγορήματος είναι:

```
?- stable(M).

M = [rick-helen, jack-lilly, john-maria, hugh-diana, greg-jenny,
     nick-sally, bill-wanda, andy-tracy, alan-patty, dick-linda]
More --> ;

M = [rick-maria, jack-lilly, john-diana, hugh-helen, greg-jenny,
     nick-sally, bill-linda, andy-tracy, alan-patty, dick-wanda]
More --> ;

M = [rick-maria, jack-lilly, john-linda, hugh-diana, greg-jenny,
     nick-sally, bill-helen, andy-tracy, alan-patty, dick-wanda]
More --> ;

M = [rick-tracy, jack-lilly, john-diana, hugh-helen, greg-jenny,
     nick-sally, bill-linda, andy-maria, alan-patty, dick-wanda]
```

Βρείτε μέχρι ποιο μέγεθος δεδομένων εισόδου το πρόγραμμά σας βρίσκει όλες τις λύσεις για το πρόβλημα σε εύλογο χρόνο. Χρησιμοποιήστε το C πρόγραμμα που θα βρείτε στο <http://www.di.uoa.gr/~takís/randstabledata.c>, το οποίο παράγει με τυχαίο τρόπο δεδομένα για το πρόβλημα (τα δεδομένα αυτά είναι κατάλληλα για την βιβλιοθήκη fd, αλλά πολύ εύκολα μπορείτε να επεκτείνετε το πρόγραμμα για να καλυφθεί και η ic).

Παραδοτέο για την άσκηση είναι ένα πηγαίο αρχείο **Prolog** με όνομα **stable.pl**.