

# Μεταπτυχιακό Πρόγραμμα Εφαρμοσμένης Πληροφορικής

## Προχωρημένη Τεχνητή Νοημοσύνη

### Εργασία 1

Προγραμματισμός με Περιορισμούς  
Χειμερινό Εξάμηνο 2017-2018

#### Άσκηση 1: Στο SuperMarket

Θέλετε να αγοράσετε προϊόντα από 7 διαφορετικές κατηγορίες (λαχανικά, απορρυπαντικά, κλπ). Το supermarket κάνει διάφορες προσφορές προϊόντων από αυτές τις ομάδες με διαφορετικές τιμές, και διαφορετική ποιότητα σε κάθε ομάδα. Για παράδειγμά σας προσφέρει προϊόντα από τις κατηγορίες {1,2,3} με τιμή 10 και δείκτη ποιότητας 5, και προϊόντα {4,7} με τιμή 5 και δείκτη ποιότητας 2. Τα δεδομένα υπάρχουν σε τρεις πίνακες

- **package**: πακέτα προσφορών ως πίνακας συνόλων ακεραίων (κατηγορίες προϊόντων),
- **package\_price**: πίνακας τιμών των αντίστοιχων πακέτων,
- **package\_quality**: πίνακας δείκτη ποιότητας των αντίστοιχων πακέτων.

Σας επιτρέπεται να πάρετε το πολύ **4 προσφορές**. Αν το συνολικό χρηματικό ποσό που μπορείτε να διαθέσετε είναι **100**, και πρέπει να αγοράσετε προϊόντα **από όλες τις κατηγορίες μόνο μια φορά**, να γράψετε ένα πρόγραμμα σε MiniZinc, το οποίο θα βρίσκει ποιες προσφορές θα πάρετε ώστε να μεγιστοποιείται η ποιότητα των αγορών σας. Το πρόγραμμα θα τυπώνει τα ακόλουθα:

```
{3,7} {4,6} {1,2,5} {}  
quality = 10;  
price = 97;
```

Τα δεδομένα βρίσκονται στο αρχείο supermarket.mzn στο compus.

#### Άσκηση 2 - Ακολουθίες

Υπάρχουν τρεις λίστες ακεραίων, όπως φαίνονται παρακάτω.

```
[2,3,4,10,22,11,17]) .  
[1,2,3,4,10,22,11,11,10,24]) .  
[2,3,4,5,10,23,10,22,11,17]) .
```

Στόχος είναι να βρεθεί η **κοινή τριάδα διαδοχικών ακεραίων** στις παραπάνω λίστες, η οποία έχει **μέγιστο άθροισμα**. Προφανώς σε κάθε λίστα η τριάδα μπορεί να βρίσκεται σε διαφορετική θέση. Στο παράδειγμά μας η τριάδα που αναζητείται φαίνεται παρακάτω.

```
[2,3,4,10,22,11,17]) .  
      | (θέση 4)  
[1,2,3,4,10,22,11,11,10,24]) .  
      | (θέση 5)  
[2,3,4,5,10,23,10,22,11,17]) .  
      | (θέση 7)
```

Να υλοποιήσετε ένα πρόγραμμα σε MiniZinc το οποίο θα επιστρέφει τις θέσεις από όπου ξεκινούν οι τριάδες που αναζητούμε σε κάθε λίστα, την τριάδα των αριθμών, και το άθροισμά της. Το αποτέλεσμα θα πρέπει να είναι στην ακόλουθη μορφή:

```
values = [10, 22, 11];
indices = [4, 5, 7];
sum = 43;
-----
=====
```

Τα δεδομένα βρίσκονται στο αρχείο align.mzn στο compus.

### Άσκηση 3 - Διανύσματα

Έστω δύο διανύσματα  $V_1$  και  $V_2$ . Το εσωτερικό γινόμενο των διανυσμάτων είναι το άθροισμα των γινομένων των αντίστοιχων συντεταγμένων τους, δηλαδή αν  $V_1 = (x_{11}, x_{12})$  και  $V_2 = (x_{21}, x_{22})$ , τότε το εσωτερικό τους γινόμενο είναι:  $V_1 \cdot V_2 = (x_{11} * x_{21} + x_{12} * x_{22})$ . Το *ελάχιστο εσωτερικό γινόμενο* ορίζεται (για τις ανάγκες της άσκησης) ως το ελάχιστο άθροισμα όρων της μορφής  $x_{1i} * x_{2j} * |i-j|$  όπου  $i, j$  είναι οι θέσεις των συντεταγμένων στα αρχικά διανύσματα, όπου κάθε συντεταγμένη χρησιμοποιείται μόνο μια φορά σε ένα όρο του αθροίσματος. Για παράδειγμα:

Διανύσματα

- (a) [2,3] και [4,5]      ελάχιστο  $2*4 * (|1-1|) + 3*5 * (|2-2|) = 0$ .  
 (b) [-2,3] και [-4,5]      ελάχιστο  $(-2)*(5) * (|1-2|) + 3 * (-4) * |2-1| = -22$

Προφανώς οι παραπάνω ορισμοί γενικεύονται σε διανύσματα  $n$ -διαστάσεων.

Να γράψετε ένα πρόγραμμα σε MiniZinc το οποίο να βρίσκει το ελάχιστο εσωτερικό γινόμενο δύο διανυσμάτων, οποιασδήποτε διάστασης. Τα διανύσματα, σας δίνονται σε αρχεία δεδομένων (**prob<x>.dzm**) της MiniZinc, μέρος του MiniZinc project που σας δίνεται.

Παράδειγμα εκτέλεσης prob1.data

```
vector1 = [2, 3];
vector2 = [4, 5];
sumTotal = 0;
-----
=====
```

Παράδειγμα εκτέλεσης prob2.data

```
vector1 = [-2, 3];
vector2 = [-4, 5];
sumTotal = -22;
-----
=====
```

Παράδειγμα εκτέλεσης prob3.data

```
vector1 = [-2, 3, 3];
vector2 = [-2, 3, 3];
sumTotal = -24;
-----
=====
```

## Άσκηση 4 - Το πρόβλημα της καλής οικοδέσποινας

Μια οικοδέσποινα οργανώνει ένα gala στο οποίο έχει προσκαλέσει  $N$  άτομα. Για να επιτύχει το gala πρέπει κάθε προσκεκλημένος να

- καθίσει ανάμεσα σε δύο άλλους του αντίθετου φύλλου (man-woman-man-woman arrangement), και
- να μοιράζεται με καθένα από αυτούς ένα *τουλάχιστον κοινό* ενδιαφέρον (hobby).

Το τραπέζι είναι *κυκλικό*, άρα οι παραπάνω κανόνες θα πρέπει να ισχύουν και για τον πρώτο και τον τελευταίο καλεσμένο. Για την επίλυση του προβλήματος σας δίνονται τα ακόλουθα στοιχεία (παράμετροι) για αριθμό προβλημάτων, στην ακόλουθη μορφή:

- $N$  είναι ο αριθμός των καλεσμένων
- **gender**, ένας μονοδιάστατος πίνακας  $N$  ακεραίων, όπου στη θέση  $i$  του πίνακα, δίνεται το φύλλο του  $i$  καλεσμένου με 1-man, 2 - woman
- **hobbies**, ένας μονοδιάστατος πίνακας  $N$  συνόλων ακεραίων, όπου στη θέση  $i$  του πίνακα, δίνονται σαν σύνολο τα ενδιαφέροντα (hobbies) του  $i$  καλεσμένου.

Για παράδειγμα:

```
N=16;
gender = [1,2,1,1,1,1,2,1,1,1,2,2,2,2,2,2];
hobbies = [ {2,1,3},
             {2,1,3},
             {3,2},
             {3,2,1},
             {2,1,3},
             {2,3,1},
             {1,2,3},
             {3,1},
             {2,3,1},
             {3,2,1},
             {1,3,2},
             {3,1,2},
             {2,3},
             {1,2},
             {2,3,1},
             {2,3} ] ;
```

Να γράψετε ένα MiniZinc πρόγραμμα που να βρίσκει την λύση στο παραπάνω πρόβλημα. Στο compus θα βρείτε όλα τα απαραίτητα αρχεία, συμπεριλαμβανομένων και των αρχείων **guests<K>.dzn** που περιέχουν δεδομένα για τα αντίστοιχα προβλήματα. Το πρόγραμμά σας θα πρέπει να δίνει τη διάταξη των καλεσμένων και το φύλλο τους, όπως φαίνεται παρακάτω:

```
guest_at = [1,16,10,15,9,14,8,13,6,12,5,11,4,7,3,2];
table_gender = [1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2];
-----
```

Η παραπάνω είναι μια από τις πολλές λύσεις που μπορεί να υπάρχουν.

β) Κάποιοι από τους καλεσμένους δεν τα πηγαίνουν καλά μεταξύ τους και άρα δεν πρέπει να καθίσουν σε διπλανές θέσεις. Οι κανόνες είναι οι ακόλουθοι:

- όσοι καλεσμένοι έχουν κάποια έχθρα με κάποιον άλλο καλεσμένο δεν πρέπει να καθίσουν στην πρώτη (1) και στην τελευταία ( $N$ ) θέση του τραπεζιού.
- δύο καλεσμένοι οι οποίοι έχουν έχθρα μεταξύ τους, δεν πρέπει να καθίσουν σε διπλανές θέσεις.

Τα νέα δεδομένα σας δίνονται ως ακολούθως:

- **pR**: ο αριθμός των ζευγών που έχουν έχθρα μεταξύ τους

- **furious**: πίνακας δύο διαστάσεων, ο οποίος δηλώνει τα ζευγάρια των "εχθρικών" μεταξύ τους καλεσμένων

Για παράδειγμα

```
pR = 3;
```

```
furious = [|16,10|13,6|12,5|];
```

σημαίνει ότι υπάρχουν τρία ζευγάρια, (16,10), (13,6) και (12,5). Σύμφωνα με τα παραπάνω οι καλεσμένοι 16 και 10, δεν πρέπει να καθίσουν σε διπλανές θέσεις.

Θα τροποποιήσετε το πρόγραμμά σας, ώστε να περιλαμβάνει και τους νέους περιορισμούς. Το

```
guest_at = [1, 13, 10, 12, 9, 7, 8, 16, 6, 15, 5, 11, 4, 2, 3, 14];
```

```
table_gender = [1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2];
```

```
Furious Placement
```

```
16 10 13 6 12 5
```

```
8 3 2 9 4 11
```

**TIP:** Η συνάρτηση **show\_int(<P><Int>)**, τυπώνει ένα ακέραιο χρησιμοποιώντας <P> θέσεις (padding). Για παράδειγμα το

```
output [show_int(4,furious[i,j]) | i in LIST, j in PAIR]
```

θα τυπώσει σε μια σειρά τους ακεραίους του πίνακα furious, σε 4 θέσεις τον καθένα.

### Σημειώσεις

- Τα απαραίτητα αρχεία για την υλοποίηση των τριών ασκήσεων θα τα βρείτε στο compus (Εγγραφα->ConstraintProgramming->CourseWork->CW1.
  - Άσκηση 1: αρχείο supermarket.mzn
  - Άσκηση 2: αρχείο allign.mzn
  - Άσκηση 3: φάκελος vectors, αρχείο project vectorsProject.mzp
  - Άσκηση 4: φάκελος gala, αρχείο project galaProject.mzp

### ΠΑΡΑΔΟΣΗ

Θα παραδώσετε εντός της ημερομηνίας που αναφέρεται στο compus τα ακόλουθα:

- Τα παραπάνω αρχεία με όνομα και τη δομή καταλόγων που έχουν στο compus **σε ένα αρχείο code.zip**.
- 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 ασκήσεις:
    - τον κώδικα σας (ασχέτως αν βρίσκεται και στο αντίστοιχο αρχείο) και σχολιασμό σχετικά με αυτόν.
    - Παραδείγματα εκτέλεσης (1-2 όπου είναι εφικτό)
    - Bugs και προβλήματα που έχει ο κώδικάς σας.

**ΠΡΟΣΟΧΉ:** ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΠΟΥ ΔΙΝΕΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ)

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