



ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (2^ο εξάμηνο)

Εργασία: week – 7 (3η Εργασία)

Ακαδημαϊκό Έτος : 2022-2023

Εισηγητής: Ηλίας Σακελλαρίου

Άσκηση 1: SuperMarket

Θέλετε να αγοράσετε προϊόντα από 7 διαφορετικές κατηγορίες (λαχανικά, απορρυπαντικά, κλπ). Το supermarket κάνει προσφορές προϊόντων από αυτές τις κατηγορίες σε ομάδες (σύνολα) 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 του προβλήματος είναι:

Ο αριθμός των συνολικών προσφορών:

```
int: ofrs = 10;
```

Το σύνολο των προσφορών. Θα παρατηρήσετε ότι έχετε 11 προσφορές συνολικά με μια “κενή” προσφορά (την προσφορά 0) με τιμή και ποιότητα ίση με 0.

```
set of int: OFFERS = 0..ofrs;
```

Το σύνολο των κατηγοριών:

```
set of int: CATEGORIES = 1..7;
```

Το σύνολο των διαφορετικών προσφορών που μπορείτε να επιλέξετε. Σας επιτρέπεται να πάρετε το πολύ 4 προσφορές:

```
set of int: BUYS = 1..4;
```

Το συνολικό χρηματικό ποσό που μπορείτε να διαθέσετε είναι 100

```
int : max_money = 100;
```

Τα δεδομένα για τις προσφορές δίνονται σε τρεις πίνακες. Ο πίνακας **package** περιέχει τα πακέτα προσφορών ως πίνακας συνόλων ακεραίων (κατηγορίες προϊόντων)

```
array[OFFERS] of set of int : package = array1d(OFFERS, [{}, {1,2,3}, ..., {2,3}]);
```

Ο πίνακας **package_price** είναι πίνακας τιμών των αντίστοιχων πακέτων:

```
array[OFFERS] of int: package_price = array1d(OFFERS,[0,10, ... ,20]);
```

και ο πίνακας **package_quality** περιέχει τον δείκτη ποιότητας των αντίστοιχων πακέτων.

```
array[OFFERS] of int: package_quality = array1d(OFFERS,[0,5, ... ,5]);
```

Όπως προαναφέρθηκε, σας επιτρέπεται να πάρετε το πολύ **4 προσφορές**. Αν το συνολικό χρηματικό ποσό που μπορείτε να διαθέσετε είναι **100** (max_money), 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:

```
%% Just for Debuging: {1,2,5} {3,6} {4,7} {}  
basket = [6, 3, 2, 0];  
quality = 6;  
price = 60;  
-----
```

Στην παραπάνω λύση επιλέγεται τις προσφορές 6, 3, 2, 0 (με την 6 να είναι {1,2,5}, κλπ) με συνολική ποιότητα των επιλεγμένων 6 και συνολική τιμή 60. Θα βρείτε τα απαραίτητα αρχεία (supermarket.mzn) στο eclass.

Σημείωση: Η λύση που φαίνεται ΔΕΝ είναι η βέλτιστη.

Άσκηση 2 - FLOWS

Κατά την σύνδεση ψυκτικών πάνελ, η εταιρία εγκατάστασης θα πρέπει να συνδέσει ένα σύνολο πανελ σε σειρά. Καθώς όμως τα τελευταία προέρχονται από διαφορετικούς κατασκευαστές, η είσοδοι και έξοδοι είναι σε διαφορετικά σημεία στο πάνελ. Ευτυχώς, η εταιρία μπορεί να χρησιμοποιήσει σωληνώσεις για να συνδέσει τις εισόδους και εξόδους (σχήμα 1 α\β). Επίσης, μπορεί να τοποθετήσει τα πανελ με στορφή 180 μοιρών, έτσι ώστε να μειώσει το μήκος της απαραίτητης καλωδίωσης (Σχήμα 1 β)).

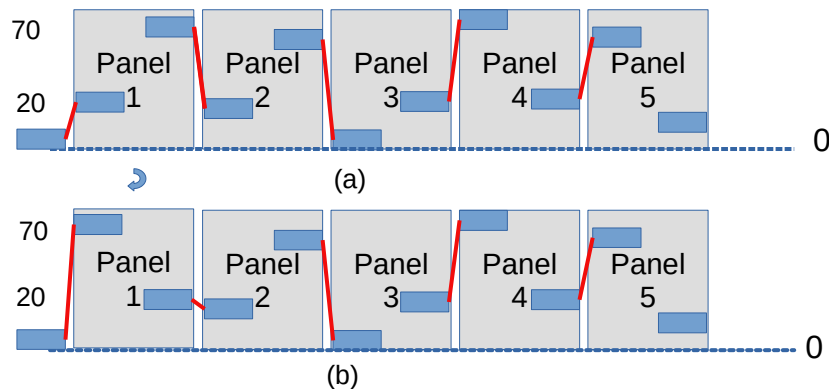


Figure 1: Συνδεσμολογία των πάνελ

Αν η θέση του κάθε πάνελ δεν μπορεί να αλλάξει, δηλαδή το Panel 1 πρέπει να μπει πρώτο, το 2 δεύτερο κ.ο.κ., το ερώτημα είναι ποιο είναι το ελάχιστο μήκος σωλήνωσης για την συνδεσμολογία. Το μήκος της σωλήνωσης προκύπτει απλά, από διαφορά στο ύψος των συνδέσεων δυο συνεχόμενων πάνελ. Για παράδειγμα στο σχήμα 1, αν υποθέσουμε ότι η είσοδος του πανελ 2 είναι σε ύψος 15, τότε με την συνδεσμολογία (α) για την σύνδεση ανάμεσα στο πανελ 1 και στο πανελ 2 απαιτούνται $70 - 15 = 55$ μονάδες σωληνώσεων, ενώ για την (β) $20 - 15 = 5$ μονάδες σωληνώσεων (λαμβάνουμε υπόψη μόνο την διαφορά ύψους). Στο ελάχιστο μήκος της συνδεσμολογίας πρέπει να περιλαμβάνεται και η αρχική σύνδεση (από το ύψος 0).

Τα δεδομένα του προβλήματος είναι τα ακόλουθα:

Ο αριθμός των συσκευών (πανελ)

```
ndevices = 3;
```

Το ύψος των συνδέσεων των συσκευών, ως πίνακας 2 διαστάσεων, με κάθε γραμμή να αφορά ένα πανελ:

```
device_levels = [|10, 40,  
|20, 60,  
|50, 60|];
```

Παράδειγμα εκτέλεσης:

```
total_length =70;  
-----
```

Θα βρείτε τα απαραίτητα αρχεία στο eclass.

Σημείωση: Η λύση που φαίνεται ΔΕΝ είναι η βέλτιστη.

Άσκηση 3 - Nuclear Reactors

Λόγω της ξηρασίας, η συνολική δυνατότητα της εταιρείας ύδρευσης σε παροχή νερού είναι πλέον 120 κυβικά μέτρα/λεπτό (κμ/λεπτό). Όμως οι πυρηνικοί σταθμοί παραγωγής ενέργειας στην περιοχή σας που πρέπει να τροφοδοτηθούν, έχουν διαφορετικές απαιτήσεις σε νερό. Οι σχετικές πληροφορίες δίνονται στα αντίστοιχα αρχεία δεδομένων (MiniZinc data files), όπως:

πόσοι πυρηνικοί αντιδραστήρες υπάρχουν:

```
no_plants = 5;
```

ποια είναι η παροχή τους σε νερό, δηλαδή πόσα κυβικά μέτρα (κμ) νερού/λεπτό μπορούν να δεχθούν:

```
supply = [50,40,60,70,90];
```

ποιες είναι οι συνολικές ανάγκες σε νερό (σε κμ)

```
needs = [100,80,180,280,270];
```

λόγω της διάταξης του δικτύου, κάποιοι αντιδραστήρες ΔΕΝ μπορούν να τροφοδοτηθούν ταυτόχρονα. Ο αριθμός των περιορισμών - προβλημάτων είναι:

```
problems = 1;
```

και το ποιοι αντιδραστήρες δεν μπορούν να τροφοδοτηθούν ταυτόχρονα, δίνεται στον ακόλουθο πίνακα 2 διαστάσεων:

```
not_sim = [|1,3|];
```

Το παραπάνω σημαίνει ότι ο αντιδραστήρας 1 και ο 3 δεν μπορούν ταυτόχρονα να δεχθούν νερό.

Προφανώς, η εταιρεία ύδρευσης μπορεί να τροφοδοτήσει περισσότερα από ένα εργοστάσια, αν η απαιτήσεις τους δεν ξεπερνούν την μέγιστη παροχή των 120 κμ/λεπτό.

Να γράψετε ένα πρόγραμμα σε MiniZinc το οποίο υπολογίζει

- **start** πότε θα ξεκινήσει η τροφοδοσία του κάθε σταθμού,
- **end_of_supply** ο χρόνος ολοκλήρωσης ολόκληρου του προγράμματος, ο οποίος θα πρέπει να είναι ο ελάχιστος δυνατός.

Για παράδειγμα (λύση δεν είναι η βέλτιστη):

```
starts = [3, 0, 0, 8, 5];  
end_of_supply = 12;
```

Όπου στο παράδειγμα ο αντιδραστήρας 1 θα ξεκινήσει την τροφοδοσία την χρονική στιγμή 3, ο 2 την χρονική στιγμή 3, κοκ. ενώ το συνολικό χρονοπρόγραμμα θα διαρκέσει συνολικά 12 λεπτά. Το παραπάνω παράδειγμα εκτέλεσης **ΔΕΝ είναι η βέλτιστη λύση**.

Θεωρείστε ότι το χρονοπρόγραμμα ξεκινά από την χρονική στιγμή 0 και πρέπει να ολοκληρωθεί σε 100 λεπτά το πολύ.

Τα απαραίτητα αρχεία θα τα βρείτε στο eclass.

Σημειώσεις

- Τα απαραίτητα αρχεία για την υλοποίηση των τριών ασκήσεων θα τα βρείτε στο ECLASS (Έγγραφα->Constraints->Assignments).
 - Άσκηση 1: φάκελος supermarket, αρχείο supermarket.mzn
 - Άσκηση 2: φάκελος Flow, αρχείο project flowProject.mzp
 - Άσκηση 3: φάκελος NuclearPlants, αρχείο project nucPlantsProject.mzp

ΠΑΡΑΔΟΣΗ

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

- Τα παραπάνω αρχεία **με όνομα και τη δομή καταλόγων** που έχουν στο ECLASS, **τα οποία θα περιέχουν τις λύσεις σας**.
- Ένα αρχείο **report.pdf (σε μορφή pdf)**, στον αρχικό φάκελο του zip, 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.

ΠΡΟΣΟΧΗ: ΝΑ ΑΚΟΛΟΥΘΗΣΕΤΕ ΑΥΣΤΗΡΑ ΤΑ ΟΝΟΜΑΤΑ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ ΠΟΥ ΔΙΝΟΝΤΑΙ ΠΑΡΑΠΑΝΩ (ΑΥΤΟΜΑΤΟΣ ΕΛΕΓΧΟΣ ΚΩΔΙΚΑ)

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