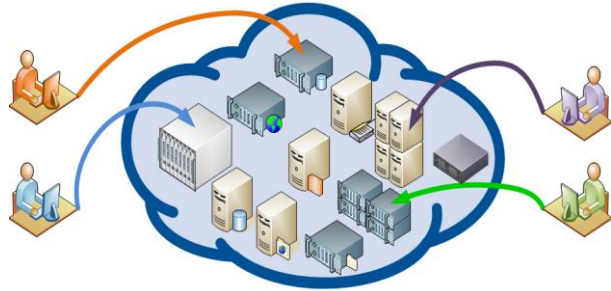


# ΑΛΓΟΡΙΘΜΟΙ 2021-2022

## Β ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ (1 βαθμός)

### Βέλτιστη ανάθεση εργασιών σε cloud μηχανήματα με χρήση δυναμικού προγραμματισμού

Οι εταιρίες παροχής υπολογιστικού νέφους (cloud computing) όπως η Amazon, διαθέτουν ένα σύνολο από εικονικές μηχανές (VMs, virtual machines) τις οποίες με τη σειρά τους, τις παρέχουν στους χρήστες έναντι συγκεκριμένης αμοιβής για το συνολικό διάστημα χρήσης. Υπάρχουν πολλοί τύποι μηχανών με διαφορετικά χαρακτηριστικά η κάθε μία, π.χ., μία εικονική μηχανή A μπορεί να υπερτερεί μίας άλλης εικονικής μηχανής ίδιου κόστους B σε εκτέλεση διεργασιών που χρησιμοποιούν περισσότερο τη CPU, ενώ η B υπερτερεί της A σε διεργασίες που προσπελαίνουν το σκληρό δίσκο. Μπορεί επίσης να υπάρχει μία ακριβότερη μηχανή Γ που να υπερτερεί τόσο της A όσο και της B στην απόδοση (αλλά κοστίζει πιο πολύ). Το πρόβλημα της επιλογής της βέλτιστης εικονικής μηχανής κάθε φορά είναι πολλές φορές δύσκολο στην πράξη. Με αυτό το πρόβλημα ασχολείται η συγκεκριμένη άσκηση.



Έχουμε μία σύνθετη διαδικασία που αποτελείται από **N βήματα** (ή διεργασίες) σε μορφή αλυσίδας:  $\Delta 1 \rightarrow \Delta 2 \rightarrow \Delta 3 \rightarrow \dots \rightarrow \Delta N$ . Τα βέλη δηλώνουν ότι τα αποτελέσματα της  $\Delta 1$  περνάνε ως είσοδος στη  $\Delta 2$ , αυτά της  $\Delta 2$  στην  $\Delta 3$ , κ.ο.κ. Έχουμε επίσης και **M εικονικούς τύπους μηχανών**. Ως είσοδο, έχουμε επίσης **δύο πίνακες**. Ο ένας είναι  $N \times M$  και δηλώνει το **συνολικό κόστος να τρέξει μία διεργασία σε ένα τύπο εικονικής μηχανής**. Ο δεύτερος πίνακας είναι  $M \times M$  και δηλώνει το **κόστος να στείλει το ένα μηχάνημα δεδομένα στο άλλο**. Ένα παράδειγμα εισόδου είναι το παρακάτω:

$N = 4$  διεργασίες ( $\Delta 1 \rightarrow \Delta 2 \rightarrow \Delta 3 \rightarrow \Delta 4$ )  
 $M = 3$  τύποι μηχανημάτων (VM1, VM2, VM3)

	VM1	VM2	VM3
$\Delta 1$	5	6	3
$\Delta 2$	7	8	5
$\Delta 3$	7	8	3
$\Delta 4$	2	7	6

	VM1	VM2	VM3
VM1	0	7	2
VM2	7	0	2
VM3	2	2	0

Στο συγκεκριμένο παράδειγμα, η σύνθετη διαδικασία έχει 4 βήματα και είναι διαθέσιμοι 3 τύποι VM. Ο πρώτος τύπος εκτελεί τις 4 διεργασίες με κόστος 5, 7, 7

και 2 αντίστοιχα. Επίσης, επικοινωνεί με τους άλλους δύο τύπους μηχανών (δεύτερο και τρίτο) με κόστος 7 και 2 αντίστοιχα. Ο δεύτερος τύπος εκτελεί τις 4 διεργασίες με κόστος 6, 8, 8 και 7 αντίστοιχα. Επίσης, επικοινωνεί με τους άλλους δύο τύπους μηχανών (πρώτο και τρίτο) με κόστος 7 και 2 αντίστοιχα. Τέλος ο τρίτος τύπος εκτελεί τις 4 διεργασίες με κόστος 3, 5, 3 και 6 αντίστοιχα. Επίσης, επικοινωνεί με τους άλλους δύο τύπους μηχανών (πρώτο και δεύτερο) με κόστος 2 και 2 αντίστοιχα. Ο πίνακας επικοινωνίας των μηχανών γενικά είναι πάντα τετραγωνικός αλλά όχι απαραίτητα συμμετρικός όπως είναι στο παραπάνω παράδειγμα.

### Τι πρέπει να υλοποιηθεί

Στην εργασία αυτή πρέπει να υλοποιήσετε έναν αλγόριθμο δυναμικού προγραμματισμού για την εύρεση του ελάχιστου κόστους της σύνθετης διαδικασίας σε κάθε βήμα και για κάθε μηχανή.

Πιο συγκεκριμένα ο αλγόριθμος αυτός θα γεμίζει έναν πίνακα **Costs N X M**, όπου κάθε κελί **Cost(i,j)** θα δείχνει το μικρότερο συνολικό κόστος της σύνθετης διαδικασίας μέχρι και το **i** βήμα όταν το **i** βήμα εκτελείται στη μηχανή **j**. Για να εκτελεστεί το βήμα **i** στη μηχανή **j**, θα πρέπει είτε τα αποτελέσματα του **i-1** βήματος να είναι ήδη στη μηχανή **j** είτε να μεταφερθούν σε αυτήν από άλλα μηχανήματα λαμβάνοντας υπόψη το κόστος επικοινωνίας. Στο παραπάνω παράδειγμα αυτός ο πίνακας θα έχει την εξής μορφή:

	VM1	VM2	VM3
Δ1	5	6	3
Δ2	12	13	8
Δ3	17	18	11
Δ4	15	20	17

Τα ελάχιστα κόστη για το πρώτο βήμα (Δ1) είναι προφανώς 5, 6, 3 ανάλογα με το αν εκτελεστεί στις VM1, VM2, VM3 αντίστοιχα (δεν έχουμε προηγούμενο βήμα).

Για το ελάχιστο κόστος μέχρι και το δεύτερο βήμα (Δ2) έχουμε:

Αν το Δ2 εκτελεστεί στην VM1 θα είναι 12 διότι έχουμε τις εξής περιπτώσεις:

→ αν το Δ1 εκτελείται στην VM1 το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης (Δ1, VM1) + Κόστος Επικοινωνίας (VM1, VM1) + Κόστος Εκτέλεσης (Δ2, VM1) =  $5 + 0 + 7 = 12$

→ αν το Δ1 εκτελείται στην VM2 το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης (Δ1, VM2) + Κόστος Επικοινωνίας (VM2, VM1) + Κόστος Εκτέλεσης (Δ2, VM1) =  $6 + 7 + 7 = 20$

→ αν το Δ1 εκτελείται στην VM3 το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης (Δ1, VM3) + Κόστος Επικοινωνίας (VM3, VM1) + Κόστος Εκτέλεσης (Δ2, VM1) =  $3 + 2 + 7 = 12$

Αν το Δ2 εκτελεστεί στην VM2 θα είναι 13 διότι έχουμε τις εξής περιπτώσεις:

→ αν το Δ1 εκτελείται στην VM1 το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης (Δ1, VM1) + Κόστος Επικοινωνίας (VM1, VM2) + Κόστος Εκτέλεσης (Δ2, VM2) =  $5 + 7 + 8 = 20$

→ αν το Δ1 εκτελείται στην VM2 το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης ( $\Delta 1, VM2$ ) + Κόστος Επικοινωνίας ( $VM2, VM2$ ) + Κόστος Εκτέλεσης ( $\Delta 2, VM2$ ) =  $6 + 0 + 8 = 14$

→ αν το  $\Delta 1$  εκτελείται στην  $VM3$  το συνολικό κόστος θα είναι:

Κόστος Εκτέλεσης ( $\Delta 1, VM3$ ) + Κόστος Επικοινωνίας ( $VM3, VM2$ ) + Κόστος Εκτέλεσης ( $\Delta 2, VM2$ ) =  $3 + 2 + 8 = 13$

κ.ο.κ.

Τελικά η λιγότερο ακριβή εκτέλεση της σύνθετης διεργασίας κοστίζει 15 κι επιτυγχάνεται όταν η τελευταία διεργασία εκτελεστεί στην πρώτη εικονική μηχανή.

Ως **είσοδος**, θα δίνεται ένα αρχείο με τα δεδομένα στην παρακάτω μορφή:

```
4
3

5 6 3
7 8 5
7 8 3
2 7 6

0 7 2
7 0 2
2 2 0
```

Δηλαδή στην πρώτη γραμμή θα δίνεται το **N**, στην δεύτερη το **M** και θα ακολουθεί μία κενή γραμμή. Στις επόμενες γραμμές θα δίνεται ο **πίνακας κόστους εκτέλεσης ( $N \times M$ )** με τα κόστη να χωρίζονται μεταξύ τους με ένα κενό. Κατόπιν ακολουθεί μία κενή γραμμή και στη συνέχεια θα δίνεται ο **πίνακας κόστους επικοινωνίας ( $M \times M$ )** με τα κόστη να χωρίζονται μεταξύ τους με ένα κενό. Όλα τα κόστη θα είναι πάντοτε **ακέρατοι και μη αρνητικοί αριθμοί**.

Το πρόγραμμα θα πρέπει να υπολογίζει και να εκτυπώνει στην οθόνη τον τελικό πίνακα  **$N \times M$**  με τα ελάχιστα κόστη αυστηρά στην εξής μορφή εξόδου:

```
5 6 3
12 13 8
17 18 11
15 20 17
```

(δηλαδή με τα κόστη να χωρίζονται μεταξύ τους με ένα κενό).

### Τρόπος υλοποίησης:

- 1) Να χρησιμοποιήσετε αποκλειστικά την γλώσσα **Java**, ακολουθώντας τις οδηγίες για την υποβολή ασκήσεων java στο σύστημα Eagle ([Οδηγίες-Eagle-2022.pdf](#)).
- 2) Το **μοναδικό όρισμα εισόδου** θα είναι το αρχείο εισόδου. Το όρισμα αυτό θα πρέπει να είναι το **args[0]** από τα ορίσματα της command line, το οποίο διαβάζεται από την **main**.
- 3) Για τη λύση να χρησιμοποιήσετε αποδοτικό αλγόριθμο δυναμικού προγραμματισμού, καθώς τα **N** και **M** μπορεί να είναι πιο μεγάλα και **θα υπάρχει χρονικός περιορισμός στην εκτέλεση των test cases**.

- 4) Ο πηγαίος κώδικας να έχει **συνοπτικά σχόλια** μέσα στον κώδικα (**inline**) και **σχόλια** επάνω από κάθε συνάρτηση, τα οποία να εξηγούν το σκεπτικό της υλοποίησής σας.

#### Παραδοτέο:

- Όλο το πρόγραμμα (πηγαίος κώδικας) θα πρέπει να είναι υλοποιημένο σε **ένα και μοναδικό αρχείο java** το οποίο θα έχει όνομα **DPnet.java** (κλάση DPnet).
- Το αρχείο αυτό θα πρέπει να το ανεβάσετε και να το **υποβάλετε** στο σύστημα **Eagle** (eagle.csd.auth.gr) με τον λογαριασμό σας.
- Στο σύστημα Eagle θα γίνεται **αυτόματη εκτέλεση** του κώδικα σας σε διάφορα test cases (και φανερά αλλά και κρυφά) καθώς και **αυτόματη βαθμολόγηση** (μέγιστος βαθμός το **100**).
- Μπορείτε να υποβάλετε στο σύστημα μέχρι και 20 φορές. Συνεπώς θα πρέπει να έχετε κάνει την υλοποίηση πρώτα σε ένα άλλο περιβάλλον της επιλογής σας (π.χ. Netbeans, IntelliJ, κλπ.) και να το έχετε δοκιμάσει ώστε **να τρέχει σωστά** και **να παράγει σωστά αποτελέσματα** πριν κάνετε υποβολές.
- Έναρξη υποβολών: **Δευτέρα 30/5/2022 10:00πμ.**
- Τέλος υποβολών: **Πέμπτη 9/6/2022 11:59μμ.**

#### Διευκρινίσεις:

- Επιτρέπεται η χρησιμοποίηση έτοιμου κώδικα ελεύθερα διαθέσιμου (έτοιμες υλοποιήσεις με κατάλληλη προσαρμογή από εσάς) με την προϋπόθεση ότι θα αναφέρεται σαφώς στα σχόλια η πηγή του. Όμως, τα προγράμματα θα ελέγχονται από αυτόματο σύστημα εντοπισμού αντιγραφών. Αν εντοπιστούν αντιγραφές μεταξύ φοιτητών, τότε οι φοιτητές αυτοί θα μηδενίζονται.
- Στο επάνω μέρος του αρχείου java με τον πηγαίο κώδικα να αναφέρονται οπωσδήποτε σε σχόλια το ονοματεπώνυμο, το ΑΕΜ και το ακαδημαϊκό email σας.
- Οι φοιτητές θα πρέπει να είναι έτοιμοι να δώσουν προφορικές εξηγήσεις για την υλοποίησή τους, εφόσον τους ζητηθεί.
- Οι βαθμοί των ασκήσεων θα ισχύουν και για τις εξετάσεις Σεπτεμβρίου ή επί πτυχίω.

**Ένα ακόμα test case:**

**Είσοδος:**

```
5
4

5 1 3 2
4 2 1 3
1 5 2 1
2 3 4 2
1 1 3 1

0 1 2 4
1 0 2 3
2 2 0 1
4 3 1 0
```

**Έξοδος:**

```
5 1 3 2
6 3 4 5
5 8 6 6
7 9 10 8
8 9 12 9
```