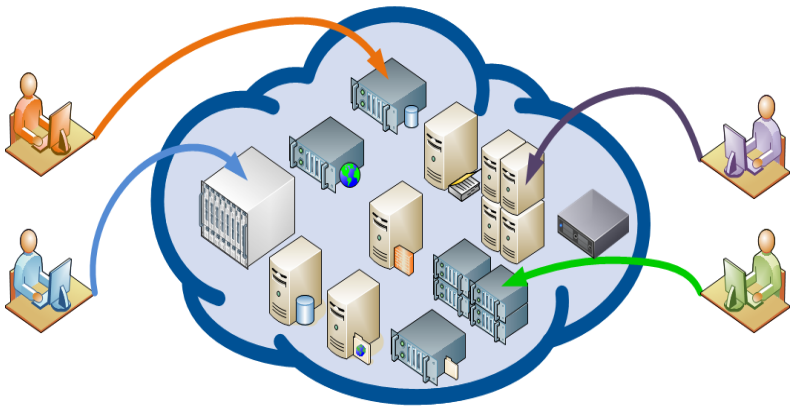
# ΑΛΓΟΡΙΘΜΟΙ 2019-2020 – 2η ΥΠΟΧΡΕΩΤΙΚΗ ΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ [1 μονάδα]



Οι εταιρίες παροχής υπολογιστικού νέφους (cloud computing) όπως η Amazon, διαθέτουν ένα σύνολο από φυσικά μηχανήματα (servers) στα οποία δημιουργούν εικονικές μηχανές (VMs, virtual machines) τις οποίες με τη σειρά τους, τις παρέχουν στους χρήστες έναντι συγκεκριμένης αμοιβής. Αυτό τους επιτρέπει από ένα φυσικό μηχάνημα να δημιουργούν πολλά περισσότερα μηχανήματα (εικονικά) τα οποία διαμοιράζονται τους πόρους του φυσικού μηχανήματος στο οποίο δημιουργήθηκαν, αυξάνοντας τα κέρδη της εταιρίας. Επιπλέον, σε κάποιες μορφές VMs, όπως π.χ., spot-instances, ο κάθε πελάτης κάνει τη δική του ξεχωριστή προσφορά για την τιμή που θα πρέπει να πληρώσει για τη μίσθωση VMs.

Ως είσοδος, θα δίνεται ένα αρχείο στην παρακάτω μορφή (χωρίς τα σχόλια):

```
180         //διαθέσιμοι πυρήνες σε όλους τους servers
16 0.23     //απαιτήσεις σε πυρήνες του 1ου πελάτη και προσφορά τιμής ανά πυρήνα
14 0.21     //απαιτήσεις σε πυρήνες του 2ου πελάτη και προσφορά τιμής ανά πυρήνα
156 0.22    //απαιτήσεις σε πυρήνες του 3ου πελάτη και προσφορά τιμής ανά πυρήνα
23 0.8      //απαιτήσεις σε πυρήνες του 4ου πελάτη και προσφορά τιμής ανά πυρήνα
5 1.1       //απαιτήσεις σε πυρήνες του 5ου πελάτη και προσφορά τιμής ανά πυρήνα
……          //κλπ. μέχρι και τον τελευταίο πελάτη.
```

Τα **πλήθη των πυρήνων** θα είναι **ακέραιοι** ενώ οι **τιμές** είναι **πραγματικοί** αριθμοί και μπορεί να έχουν **το πολύ έως 3 δεκαδικά ψηφία**. Ζητείται να υλοποιηθεί ένα πρόγραμμα που να επιτελεί τις ακόλουθες λειτουργίες (πρώτα την Α και αμέσως μετά την Β):

**ΛΕΙΤΟΥΡΓΙΑ Α**

Ο πάροχος, για να μειώσει το φόρτο διαχείρισης, επιθυμεί να έχει το μικρότερο δυνατό αριθμό VMs ανά πελάτη. Οπότε, ζητείται να βρεθεί **το ελάχιστο πλήθος από VMs ανά πελάτη**, υποθέτοντας ότι ο πάροχος μπορεί να σηκώνει VMs **με αποκλειστικά 1, 2, 7 ή 11 πυρήνες**. Το πρόγραμμα θα πρέπει να εκτυπώνει στην οθόνη το ελάχιστο πλήθος ανά πελάτη αυστηρά στην εξής μορφή εξόδου:

```
Client 1: 3 VMs
Client 2: 2 VMs
Client 3: …
```

(Πληροφοριακά, το ελάχιστο πλήθος στον 1ο πελάτη προκύπτει από 2 VM 7 πυρήνων και 1 VM 2 πυρήνων. Το ελάχιστο πλήθος στον 2ο πελάτη προκύπτει από 2 VM 7 πυρήνων.)

**ΛΕΙΤΟΥΡΓΙΑ Β**

Ο πάροχος δεν έχει αρκετούς φυσικούς πυρήνες ώστε να ικανοποιήσει όλους τους πελάτες, ενώ απαγορεύεται πολλά VMs να μοιράζονται τον ίδιο φυσικό πυρήνα. Με βάση τους **διαθέσιμους πυρήνες** ζητείται να βρεθεί το **μέγιστο ποσό πληρωμής** που μπορεί να προκύψει (επιλέγοντας τους κατάλληλους πελάτες). Το πρόγραμμα θα πρέπει να εκτυπώνει στην οθόνη το μέγιστο αυτό ποσό αυστηρά στην εξής μορφή εξόδου (σε νέα γραμμή και αμέσως μετά την έξοδο της λειτουργίας Α):

```
Total amount: 234.32
```

**Τρόπος υλοποίησης:**

1) Να χρησιμοποιήσετε αποκλειστικά την γλώσσα **Java.**
2) Το **μοναδικό όρισμα εισόδου** να είναι το αρχείο εισόδου (το όνομά του). 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 αλγορίθμους που έχετε διδαχθεί. Θα πρέπει εσείς να εντοπίσετε ποιοι αλγόριθμοι θα πρέπει να χρησιμοποιηθούν από αυτούς που περιέχονται στην ύλη του μαθήματος. Οι αλγόριθμοι θα πρέπει να είναι αποδοτικοί καθώς θα υπάρχει χρονικός περιορισμός στην εκτέλεση των test cases (να μην χρησιμοποιηθεί ωμή βία).
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 Cores.java** (κλάση Cores).
- 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 κώδικά 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 χωρίς ποινή αλλά μετά την 20η φορά θα υπάρχει απώλεια βαθμού κατά 1%. 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 25/5/2020.**
- **Τέλος υποβολών: Κυριακή 7/6/2020 11:59μμ.**

**Διευκρινήσεις:**

- Επιτρέπεται η χρησιμοποίηση έτοιμου κώδικα από τρίτους με την προϋπόθεση ότι θα αναφέρεται σαφώς στα σχόλια. Όμως, όπως αναγράφεται στην ιστοσελίδα του μαθήματος, τα προγράμματα θα ελέγχονται από αυτόματο σύστημα εντοπισμού αντιγραφών (αυτό που υπάρχει στο Eagle). Αν εντοπιστούν αντιγραφές μεταξύ φοιτητών, τότε οι φοιτητές θα μηδενίζονται συνολικά στο μάθημα.
- 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 μέχρι να ξαναδιδαχθεί το μάθημα.***

* μορφή αρχείου πηγαίου κώδικα (**`Cores.java`**):

```
import …….;
import …….;
……
public class Cores {
    ……
    ……
   public static void main(String[] args)
   {
            //στο args[0] θα περνάει το όνομα του αρχείου εισόδου με τα δεδομένα
            ……
            ……
   }
}
```