

# ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

---

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ

---

## Δομές Δεδομένων Εργασία II

---

Αναφορά Εργασίας

Ονοματεπώνυμο: Ελένη Δορκοφίκη - Γεωργία Καραγεώργου  
Αριθμό Μητρώου: p3220046 - s6200039  
Έτος σπουδών: 3ο - 5ο

## 1 Χρήση της ουράς προτεραιότητας και γενική περιγραφή του προγράμματος

Η ουρά προτεραιότητας χρησιμοποιείται στην υλοποίηση του αλγορίθμου **Greedy** προκειμένου να διαχειριστούμε τους επεξεργαστές με βάση τον συνολικό τους χρόνο επεξεργασίας. Για την αποθήκευση των επεξεργαστών, υλοποιήθηκε μία μέγιστη ουρά προτεραιότητας (**MaxPQ**) η οποία επιτρέπει την εύκολη ανάκτηση του επεξεργαστή με τον μικρότερο συνολικό φόρτο εργασίας.

**Τα κύρια βήματα της υλοποίησης έχουν ως εξής:**

1. **Διαβάζουμε τα δεδομένα εισόδου** από το αρχείο, όπου περιλαμβάνονται ο αριθμός των επεξεργαστών και των εργασιών, καθώς και οι χρόνοι εκτέλεσης κάθε εργασίας
2. **Αποθηκεύουμε τις εργασίες σε μία λίστα** ( $List<Job>$ ), όπου κάθε αντικείμενο  $Job$  αναπαριστά μία εργασία με τα αντίστοιχα χαρακτηριστικά του.
3. **Αρχικοποιούμε τους επεξεργαστές** δημιουργώντας αντικείμενα  $Processor$  και τα αποθηκεύουμε στην ουρά προτεραιότητας (**MaxPQ**)
4. **Εκτελούμε τον αλγόριθμο Greedy:**
  - Για κάθε εργασία, βρίσκουμε τον επεξεργαστή με τον μικρότερο τρέχοντα φόρτο εργασίας.
  - Εκχωρούμε την εργασία σε αυτόν τον επεξεργαστή και ενημερώνουμε την ουρά προτεραιότητας.
5. **Υπολογίζουμε το Makespan**, δηλαδή τον μέγιστο χρόνο που απαιτείται για την ολοκλήρωση όλων των εργασιών.

Η έξοδος του αλγορίθμου είναι η τιμή του makespan, η οποία καταγράφεται και συγκρίνεται για διαφορετικά δεδομένα εισόδου.

## 2 Αλγόριθμος ταξινόμησης

Ο αλγόριθμος ταξινόμησης που επιλέχθηκε είναι **Quicksort** και εφαρμόζεται πάνω στη λίστα των εργασιών ( $List<Job>$ ).

Η ταξινόμηση πραγματοποιείται με βάση το χρόνο εκτέλεσης κάθε εργασίας σε φθίνουσα σειρά, ώστε οι μεγαλύτερες εργασίες να τοποθετηθούν πρώτες στους επεξεργαστές. Αυτή η προσέγγιση βελτιώνει την κατανομή του φόρτου εργασίας και ενδέχεται να μειώσει το makespan.

### 3 Συμπεράσματα από την πειραματική αξιολόγηση

Η πειραματική αξιολόγηση πραγματοποιήθηκε με διαφορετικά μεγέθη  $N(100, 250, 500)$ . Για κάθε  $N$ , εκτελέστηκαν 10 δοκιμές και καταγράφηκαν οι μέσες τιμές του makespan για τις δύο εκδοχές του αλγορίθμου (Greedy και SortedGreedy).

N	Average Makespan Greedy	Average Makespan SortedGreedy
100	541.1	503.6
250	882.4	834.1
500	1212.4	1161.0

Table 1: Results

Όπως παρατηρούμε, η ταξινόμηση των εργασιών πριν την εκχώρησή τους βελτιώνει την κατανομή του φόρτου εργασίας, οδηγώντας σε μικρότερο makespan στις περισσότερες περιπτώσεις. Η στρατηγική αυτή συμβάλλει στη μείωση της ανισοκατανομής εργασιών μεταξύ των επεξεργαστών. .

**Σημείωση:** Λόγω τυχαιότητας τα αποτελέσματα είναι διαφορετικά σε κάθε εκτέλεση, ωστόσο κυμαίνονται περίπου στο ίδιο εύρος.

### 4 Οδηγίες εκτέλεσης

Το πρόγραμμα εκτελείται με main την Comparisons , κατά την εκτέλεση το output θα εμφανιστεί όλα τα αποτελέσματα και τις συγκρίσεις στη κονσόλα (η υλοποίηση μας γίνεται μέσω των λογισμικών Eclipse/IntelJ)

Οι είσοδοι είναι αρχεία μορφής .txt , όπου αποθηκεύονται οι εργασίες και οι επεξεργαστές και δημιουργούνται στην Comparisons, ενώ τα αποτελέσματα εμφανίζονται στο τερματικό.