



# Άσκηση 1

ΠΑΡΑΛΛΗΛΑ ΣΥΣΤΗΜΑΤΑ

Οικονόμου Βασίλειος | Α.Μ.: 71347451

## Δομή του προγράμματος

Για το πρόγραμμα έχουν οριστεί τύποι δεδομένων και συναρτήσεις

- ❖ Τύποι δεδομένων: Η δομή Array1D χρησιμοποιείται για την σύνθεση της Array2D, όπως φαίνεται παρακάτω. Με την δομή Array2D αναπαριστούνται τα δεδομένα του δοθέντος δισδιάστατου πίνακα της και του ζητούμενου πίνακα B.

- Array1D

```
typedef struct Array1D
{
    int columnLen;
    int *column;
}Array1D;
```

- Array2D

```
typedef struct Array2D
{
    int rowLen;
    Array1D *row;
}Array2D;
```

- ❖ Συναρτήσεις:

- **void initArray2D(Array2D \*array)**: Κάνει αρχικοποίησή έναν πίνακα τύπου Array2D
- **void addRow(Array2D \*array)**: Προσθέτει μία γραμμή σε έναν πίνακα τύπου Array2D δεσμεύοντας περισσότερη μνήμη στο Array1D, όπου περιέχει ο τύπος.
- **void addColumn(Array1D \*array)**: Προσθέτει μία στήλη σε έναν πίνακα τύπου Array1D δεσμεύοντας περισσότερη μνήμη στον πίνακα column, όπου περιέχει ο τύπος.
- **void insertColumnValue(Array2D \*array, int rowIndex, int value)**: προσθέτει μία στήλη και τιμή σε μία δεδομένη γραμμή του πίνακα. Εάν η γραμμή δεν υπάρχει και είναι η επόμενη του πίνακα, τότε την δημιουργεί.
- **Array2D readArray2D(char \*filePath)**: Εισάγει έναν δισδιάστατο πίνακα από αρχείο.
- **void displayArray2D(Array2D array)**: Τυπώνει στο τερματικό τα δεδομένα τύπου δεδομένου Array2D
- **void freeArray2D(Array2D \*array)**: Αποδεσμεύει τον χώρο των δεικτών του τύπου δεδομένου Array2D
- **int isSymetrical(Array2D array)**: Ελέγχει εάν ένας πίνακας είναι συμμετρικός (τετράγωνος)
- **int isDiagonallyDominant(Array2D array)**: Ελέγχει παράλληλα εάν ο πίνακας είναι διαγώνια δεσπόζων με την χρήση των threads.
- **int maxValue(Array2D array)**: Βρίσκει την μέγιστη τιμή του πίνακα με την χρήση των threads.
- **void createSymmetricalSize(Array2D \*targetArray, int dimension)**: Δημιουργεί συμμετρικό (ή τετραγωνικό) πίνακα με βάση τις διαστάσεις άλλου πίνακα.

- **Array2D buildArray(const Array2D inputArray, int max):** Δημιουργεί το πίνακα του ερωτήματος παράλληλα με την χρήση των threads.
- **int minValue\_Reduction(Array2D array):** Βρίσκει την ελάχιστη τιμή του πίνακα με την χρήση των threads με το reduction clause.
- **int minValue\_Critical(Array2D array):** Βρίσκει την ελάχιστη τιμή του πίνακα με την χρήση των threads χρησιμοποιώντας την κρίσιμη περιοχή.
- **int minValue\_BinaryTree(Array2D array):** Βρίσκει την ελάχιστη τιμή του πίνακα με την χρήση των threads και του δυαδικού δέντρου.

❖ Σταθερές:

- **FILEPATH:** Περιέχει το μονοπάτι του αρχείου, που θα διαβάσει τον πίνακα εισόδου.
- **SYMMETRICAL:** Περιέχει την τιμή επιστροφής της συνάρτησης isSymetrical, στην περίπτωση που ο πίνακας είναι συμμετρικός
- **ASYMMETRICAL:** Περιέχει την τιμή επιστροφής της συνάρτησης isSymetrical, στην περίπτωση που ο πίνακας είναι μη συμμετρικός.
- **DIAGONALLY\_DOMINANT:** Η τιμή επιστροφής της συνάρτησης isDiagonallyDominant στην περίπτωση που είναι διαγώνια δεσπόζων.
- **NOT\_DIAGONALLY\_DOMINANT:** Η τιμή επιστροφής της συνάρτησης isDiagonallyDominant στην περίπτωση που δεν είναι διαγώνια δεσπόζων.

## Επεξήγηση του προγράμματος

Η ρουτίνα του προγράμματος είναι η εξής:

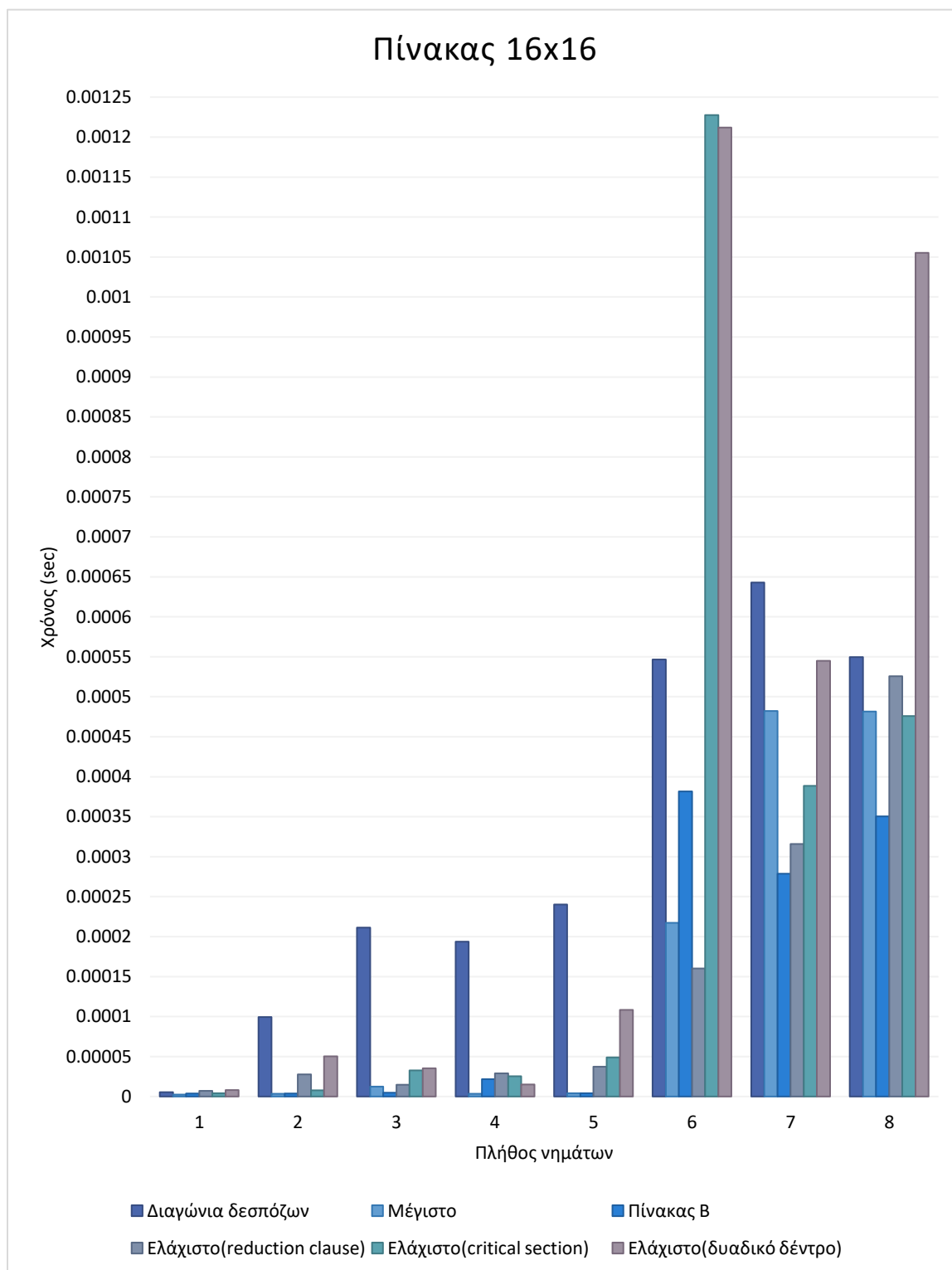
1. καλεί την συνάρτηση readArray2D για να διαβάσει τον πίνακα από το δηλωμένο μονοπάτι του αρχείου.
2. Ελέγχει εάν είναι συμμετρικός ο πίνακας καλώντας την συνάρτηση isSymetrical.
3. Ελέγχει εάν είναι διαγώνια δεσπόζων καλώντας την συνάρτηση isDiagonallyDominant.
4. Υπολογίζει την μέγιστη τιμή του πίνακα καλώντας την συνάρτηση maxValue.
5. Δημιουργεί τον πίνακα B καλώντας την συνάρτηση buildArray.
6. Βρίσκει την ελάχιστη τιμή με το reduction clause καλώντας την συνάρτηση minValue\_Reduction.
7. Βρίσκει την ελάχιστη τιμή με την κρίσιμη περιοχή καλώντας την συνάρτηση minValue\_Critical.
8. Βρίσκει την ελάχιστη τιμή με ελάττωση της τιμής με δυαδικό δέντρο καλώντας την συνάρτηση minValue\_BinaryTree.

## Αποτελέσματα και χρόνοι του προγράμματος σε διάφορες συνθήκες

Για τα αποτελέσματα το πρόγραμμα έτρεξε για πέντε φορές για κάθε συνδυασμό μεγέθους πίνακα-πλήθος νημάτων και υπολογίστηκε ο μέσος όρος του χρόνου τρεξίματος. Ο χρόνος υπολογίστηκε με την χρήση της συνάρτησης omp\_get\_wtime του openMP.

**Παρατηρήσεις:** Το πρόγραμμα για μερικές τιμές των threads έκανε περισσότερο χρόνο από το αναμενόμενο.

Παρακάτω είναι τα σχετικά διαγράμματα των χρόνων τρεξίματος.



Πίνακας 32x32

