

# Εργαστήριο 4

(αλγόριθμοι αναζήτησης)

1. Υλοποιήστε σαν συνάρτηση τον αλγόριθμο δυαδικής αναζήτησης (τον επαναληπτικό και ΟΧΙ τον αναδρομικό).

**int binarySearch (int pin[], int l, int r, int num)**

2. Υλοποιήστε σαν συνάρτηση τον αλγόριθμο αναζήτησης παρεμβολής.

**int interpolationSearch (int pin[], int low, int high, int num)**

Και οι δύο αυτοί αλγόριθμοι δίνονται τόσο στις σημειώσεις όσο και στις διαλέξεις του μαθήματος.

3. Συγκρίνετε τον αριθμό των βημάτων που εκτελεί ο καθένας από αυτούς τους αλγορίθμους για να εντοπίσει τη θέση στοιχείου το οποίο υπάρχει ήδη σε πίνακα με τυχαίες (random) τιμές οι οποίες είναι σε αύξουσα διάταξη.

Όταν λέμε αριθμό βημάτων εννοούμε τον αριθμό των επαναλήψεων που εκτελεί ο βασικός βρόγχος (εξωτερικό while loop) του κάθε αλγορίθμου.

Το στοιχείο που θα αναζητάει ο αλγόριθμος θα πρέπει να είναι ήδη μέσα στον πίνακα. Δηλαδή όταν καλείτε τις συναρτήσεις binarySearch και interpolationSearch, στο όρισμα num θα περνάτε ένα τυχαίο στοιχείο που υπάρχει ήδη στον πίνακα.

Ένα κομμάτι κώδικα με το οποίο μπορείτε να γεμίσετε έναν πίνακα με τυχαίες τιμές σε αύξουσα διάταξη είναι:

```
int pin[N], i;

/* Initialize array */
srand(time(NULL));
pin[0] = rand()%100;
for(i=1;i<N;i++)
    pin[i] = pin[i-1] + rand()%10;
```

**Θα πρέπει να παραδώσετε:**

- Τους κώδικες των δύο αλγορίθμων
- Μία μικρή αναφορά όπου θα αναφέρετε τον αριθμό βημάτων για κάθε αλγόριθμο και για διαφορετικά πλήθη των στοιχείων εισόδου, δηλαδή για 100, 1000 και 5000 στοιχεία στον πίνακα. Για κάθε διαφορετικό πλήθος εισόδου θα πρέπει να τρέξετε τον κάθε αλγόριθμο πέντε φορές.