

**Εργαστήριο Δομών Δεδομένων - 1<sup>η</sup> Άσκηση**  
**Ημερομηνία Παράδοσης : 26/10/2016 (πριν την έναρξη του εργαστηρίου – 11.00 -)**

**Εργαστηριακός Διδάσκων Μαθήματος:** Δούμα Αναστασία (sia@aegean.gr)

Στη πρώτη εργαστηριακή άσκηση καλείστε να ασχοληθείτε με τη σειριακή και τη δυαδική αναζήτηση και να προσδιορίσετε πειραματικά τον χρόνο εκτέλεσης τους.

**Ζήτημα 1 (Βαρύτητα 45%):**

Να υλοποιήσετε τον αλγόριθμο σειριακής αναζήτησης και να τον εφαρμόσετε σε πίνακα  $N$  στοιχείων. Το πρόγραμμα αρχικά θα διαβάζει το μέγεθος  $N$  του πίνακα και θα δεσμεύει την αντίστοιχη μνήμη. Ο πίνακας θα αρχικοποιείται με  $N$  τυχαίους ακέραιους αριθμούς από το διάστημα  $[1, 30.000]$ . Στη συνέχεια, το πρόγραμμα θα εκτελεί 1000 φορές τα εξής:

- Επιλογή τυχαίου αριθμού  $r$  από το διάστημα  $[1, 30.000]$ .
- Σειριακή αναζήτηση του αριθμού  $r$  στον πίνακα.

Το πρόγραμμα θα πρέπει να εμφανίζει το μέσο αριθμό συγκρίσεων μεταξύ στοιχείων του πίνακα και το μέσο χρόνο για κάθε σειριακή αναζήτηση. Επίσης το πρόγραμμα θα εμφανίζει και το ποσοστό των τυχαίων αριθμών που βρέθηκαν στον πίνακα δηλαδή το ποσοστό των επιτυχημένων αναζητήσεων.

**Ζητούμενα:**

(α) Να εκτελέσετε το πρόγραμμα για  $N$  (διάσταση πίνακα) = 5.000, 20.000, 35.000, 50.000 και 65.000 και να συμπληρώσετε τον παρακάτω πίνακα:

N	Μέσος #συγκρίσεων	Μέσος χρόνος αναζήτησης	Ποσοστό επιτυχημένων αναζητήσεων
5.000			
20.000			
35.000			
50.000			
65.000			

(β) Να κάνετε τις γραφικές παραστάσεις του μέσου αριθμού συγκρίσεων και του μέσου χρόνου αναζήτησης σαν συνάρτηση του αριθμού των στοιχείων  $N$  (χρησιμοποιείτε excel ή κάποιο άλλο σχετικό λογισμικό).

(γ) Πώς μεταβάλλεται ο μέσος αριθμός συγκρίσεων και ο μέσος χρόνος σειριακής αναζήτησης σαν συνάρτηση του  $N$  (π.χ. σαν λογαριθμική συνάρτηση, γραμμική συνάρτηση, τετραγωνική συνάρτηση). Συμφωνούν τα αποτελέσματά σας με τη θεωρητική ανάλυση;

### Ζήτημα 2 (Βαρύτητα 45%):

Να υλοποιήσετε τον αλγόριθμο δυαδικής αναζήτησης και να τον εφαρμόσετε σε πίνακα  $A$  με  $N$  στοιχεία. Το πρόγραμμα αρχικά θα διαβάσει το μέγεθος  $N$  του πίνακα και θα δεσμεύει την αντίστοιχη μνήμη. Ο πίνακας θα αρχικοποιείται με  $N$  ταξινομημένους ακέραιους αριθμούς.

Πιο συγκεκριμένα, να θέσετε  $A[0] = 1 + \text{rand}() \% 3$ , και για κάθε  $i=1, \dots, N-1$   
 $A[i] = A[i-1] + 1 + \text{rand}() \% 3$ .

Στη συνέχεια, το πρόγραμμα θα εκτελεί 1000 φορές τα εξής:

- Επιλογή τυχαίου αριθμού  $r$  από το διάστημα που ορίζετε με βάση τις τιμές του πίνακα.
- Δυαδική αναζήτηση του αριθμού  $r$  στον πίνακα  $A$ .

Το πρόγραμμα θα πρέπει να εμφανίζει το μέσο αριθμό συγκρίσεων μεταξύ στοιχείων του πίνακα και το μέσο χρόνο για κάθε δυαδική αναζήτηση. Επίσης το πρόγραμμα θα εμφανίζει και το ποσοστό των τυχαίων αριθμών που βρέθηκαν στον πίνακα δηλαδή το ποσοστό των επιτυχημένων αναζητήσεων.

#### Ζητούμενα:

(α) Να εκτελέσετε το πρόγραμμα για  $N$  (διάσταση πίνακα) = 5.000, 20.000, 35.000, 50.000 και 65.000 και να συμπληρώσετε τον παρακάτω πίνακα:

N	Μέσος #συγκρίσεων	Μέσος χρόνος αναζήτησης	Ποσοστό επιτυχημένων αναζητήσεων
5.000			
20.000			
35.000			
50.000			
65.000			

(β) Να κάνετε τις γραφικές παραστάσεις του μέσου αριθμού συγκρίσεων και του μέσου χρόνου αναζήτησης σαν συνάρτηση του αριθμού των στοιχείων  $N$ .

(γ) Πώς μεταβάλλεται ο μέσος αριθμός συγκρίσεων και ο μέσος χρόνος αναζήτησης σαν συνάρτηση του  $N$  (π.χ. σαν λογαριθμική συνάρτηση, γραμμική συνάρτηση, τετραγωνική συνάρτηση). Συμφωνούν τα αποτελέσματα σας με τη θεωρητική ανάλυση;

(δ) Να συγκρίνετε το χρόνο εκτέλεσης των αλγόριθμων σειριακής και δυαδικής αναζήτησης με βάση τα αποτελέσματα των ζητημάτων 1 και 2. Τι συμπεράσματα βγάξετε;

### Ζήτημα 3 (Βαρύτητα 10%):

Έστω οι ακόλουθες συναρτήσεις:

<b>α)</b> <pre>int fun1(int N) {     int x = 0;     for (int i = 0; i &lt; N; i++)         x++;      return x; }</pre>	<b>β)</b> <pre>int fun2(int N) {     if (N == 1) return 0;     return 1 + fun2(N/2); }</pre>
<b>γ)</b> <pre>int fun3(int N) {     int x = 0;     for (int i = 0; i &lt; N; i++)         for (int j = 0; j &lt; i; j++)             x++;     return x; }</pre>	<b>δ)</b> <pre>int fun4(int N) {     int x;      x=N+2;     return x; }</pre>

Αν υποθέσουμε ότι οι συναρτήσεις καλούνται με όρισμα θετικό ακέραιο  $N > 0$ , ποια είναι η πολυπλοκότητα χρόνου συναρτήσεως του  $N$  για κάθε μια από τις συναρτήσεις; Δικαιολογήστε την απάντησή σας.

**Σημείωση** (για τα 2 πρώτα ζητήματα) :

- Υλοποιείτε ένα πρόγραμμα το οποίο θα περιλαμβάνει και τις δύο υλοποιήσεις των αλγορίθμων αναζήτησης. Ο χρήστης με κατάλληλο μενού θα επιλέγει ποιον αλγόριθμο θέλει να εκτελέσει κάθε φορά.
- Οι αλγόριθμοι της σειριακής και της δυαδικής αναζήτησης **θα πρέπει να υλοποιηθούν σε σχετικές συναρτήσεις**, οι οποίες θα καλούνται κάθε φορά από το κυρίως πρόγραμμα. Το ποια θα είναι η υπογραφή της κάθε συνάρτησης το αποφασίζετε εσείς ανάλογα με την υλοποίησή σας.
- Όλες οι υλοποιήσεις να γίνουν με **δυναμική δέσμευση μνήμης**.
- Απαιτείται η **εισαγωγή «έξυπνων» σχολίων** στον κώδικά σας.

#### Παραδοτέα:

α) Τον κώδικα που θα έχετε υλοποιήσει μέσα στο εργαστηρίου θα πρέπει να τον παραδώσετε με την ολοκλήρωση του εργαστηρίου (ισχύει μόνο για όσους παρακολουθούν το εργαστήριο). Υπάρχει σχετικός σύνδεσμος στο eclass. Το αρχείο που θα ανεβάσετε θα πρέπει να έχει όνομα **Exercise1\_lab.cpp** (ή **.c** - **θα πρέπει να το κάνετε .zip για να το ανεβάσετε**). Στην 1<sup>η</sup> γραμμή του κώδικά σας θα πρέπει υποχρεωτικά να αναγράφεται το όνομα και ο αριθμός μητρώου των μελών της ομάδας που συμμετείχε στην υλοποίηση που έγινε στο εργαστήριο.

(β) Τελική παράδοση εργασίας :

Για την τελική παράδοση της εργασίας θα πρέπει να δημιουργήσετε **ένα αρχείο** με τον πηγαίο κώδικα που θα περιέχει και τα 2 ζητήματα με προέκταση **.c** ή **cpp** και όνομα **Exercise1.cpp** (ή **Exercise1.c**). Τον πίνακα συμπληρωμένο, τις γραφικές παραστάσεις και τις

απαντήσεις στο 3<sup>ο</sup> ζήτημα θα πρέπει να τις παραδώσετε σε ξεχωριστό αρχείο με όνομα **Exercise1.pdf**.

**Τέλος θα δημιουργήσετε ένα zip αρχείο με όνομα αρχείου «το email σας\_όνομα άσκησης» (πχ. icsd000666\_Exercise1.zip) (μόνο ένα μέλος της ομάδας θα πρέπει να καταγράψει το email του). Το zip αρχείο θα πρέπει να περιέχει τα παραπάνω δύο αρχεία. ΠΡΟΣΟΧΗ: ΚΑΜΙΑ ΕΡΓΑΣΙΑ ΔΕΝ ΘΑ ΔΙΟΡΘΩΘΕΙ ΑΝ ΔΕΝ ΑΠΟΣΤΑΛΕΙ ΜΕ ΤΟΝ ΠΑΡΑΠΑΝΩ ΤΡΟΠΟ!**

Η παράδοση του αρχείου θα γίνει με τη χρήση της πλατφόρμας ηλεκτρονικής μάθησης του τμήματος (<http://www.icsd.aegean.gr/eclass>).

Σε όλα τα αρχεία που θα δημιουργήσετε (πηγαίος κώδικας και pdf) θα πρέπει να αναφέρετε στην αρχή του κειμένου τα ονόματα των μελών της ομάδας εργασίας.

**Υποδείξεις.** Η εργασία είναι **2 ατόμων**. Τα προγράμματα πρέπει να υλοποιηθούν σε γλώσσα C ή C++.

**Μπορείτε να παράγετε τυχαίους αριθμούς με τον παρακάτω κώδικα.** Η αρχικοποίηση της γεννήτριας τυχαίων αριθμών μπορεί να γίνει με τον Αριθμό Μητρώου σας ή να χρησιμοποιήσετε την ώρα του συστήματος για να έχετε παραγωγή διαφορετικών τυχαίων αριθμών σε κάθε εκτέλεση.

```
#include <stdlib.h>
#define LIMIT 30000

srand((long) 2000666); /* Αρχικοποίηση γεννήτριας τυχαίων αριθμών με AM */
int x = rand() % LIMIT; /* οι τυχαίοι αριθμοί στο διάστημα [0,LIMIT] */
```

**Μπορείτε να μετρήσετε το χρόνο εκτέλεσης ενός τμήματος κώδικα ως εξής:**

```
#include <time.h>
clock_t start, end;
double elapsed;
start = clock();

/* Κώδικας του οποίου το χρόνο εκτέλεσης θέλουμε να μετρήσουμε */

end = clock();
elapsed = ((double) (end - start)) / CLOCKS_PER_SEC;
```