



Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

Σχολή Θετικών Επιστημών, Πανεπιστήμιο Αιγαίου

Εργαστήριο Δομών Δεδομένων – 1^η Άσκηση

Διδάσκοντες: Θεόδωρος Ευδωρίδης, Υποψήφιος Διδάκτορας
Δημήτρης Φωτάκης, Επίκουρος Καθηγητής

Ονοματεπώνυμο Φοιτητή:

Α.Μ.:

Εξάμηνο:

Στην πρώτη εργαστηριακή άσκηση καλείστε να προσδιορίσετε πειραματικά, το χρόνο εκτέλεσης της σειριακής και της δυαδικής αναζήτησης.

Ζήτημα 1. 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 $[1, 30000]$. Στη συνέχεια, το πρόγραμμα θα εκτελεί 1000 φορές τα εξής:

- Επιλογή τυχαίου αριθμού r από το διάστημα $[1, 30000]$.
- Σειριακή αναζήτηση του αριθμού r στον πίνακα.

Τέλος, το πρόγραμμα θα επιστρέφει το μέσο αριθμό συγκρίσεων μεταξύ στοιχείων του πίνακα και το μέσο χρόνο για κάθε σειριακή αναζήτηση. Επίσης το πρόγραμμα θα επιστρέφει το ποσοστό των τυχαίων αριθμών που βρέθηκαν στον πίνακα (ποσοστό επιτυχημένων αναζητήσεων).

Ζητούμενα: (α) Να εκτελέσετε το πρόγραμμα για $n = 5000, 10000, 20000$, και 50000 και να συμπληρώσετε τον παρακάτω πίνακα:

n	Μέσος #συγκρίσεων	Μέσος χρόνος αναζήτησης	Ποσοστό επιτ. αναζητήσεων
5000			
10000			
20000			
50000			

(β) Να κάνετε επίσης 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 σειριακής αναζήτησης σαν συνάρτηση του n (π.χ. σαν λογαριθμική συνάρτηση, γραμμική συνάρτηση, τετραγωνική συνάρτηση). Συμφωνούν τα αποτελέσματά σας με τη θεωρητική ανάλυση;

Χώρος απάντησης:

Ζήτημα 2. Να υλοποιήσετε τον αλγόριθμο *δυναδικής αναζήτησης* και να τον εφαρμόσετε σε πίνακα A με n στοιχεία. Το πρόγραμμα αρχικά θα διαβάζει το μέγεθος n του πίνακα και θα δεσμεύει την αντίστοιχη μνήμη. Ο πίνακας θα αρχικοποιείται με n ταξινομημένους ακέραιους αριθμούς στο διάστημα $[1, 30000]$. 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:

3. Επιλογή τυχαίου αριθμού r από το διάστημα $[1, 30000]$.
4. Δυναδική αναζήτηση του αριθμού r στον πίνακα A .

Τέλος, το πρόγραμμα θα επιστρέφει το μέσο αριθμό συγκρίσεων μεταξύ στοιχείων του πίνακα και το μέσο χρόνο για κάθε δυναδική αναζήτηση. Επίσης το πρόγραμμα θα επιστρέφει το ποσοστό των τυχαίων αριθμών που βρέθηκαν στον πίνακα (ποσοστό επιτυχημένων αναζητήσεων).

Ζητούμενα: (α) Να εκτελέσετε το πρόγραμμα για $n = 5000, 10000, 20000$, και 50000 και να συμπληρώσετε τον παρακάτω πίνακα:

n	Μέσος #συγκρίσεων	Μέσος χρόνος αναζήτησης	Ποσοστό επιτ. αναζητήσεων
5000			
10000			
20000			
50000			

(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).

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

Χώρος Απάντησης:

Παραδοτέα: (α) Ο πηγαίος κώδικας σε δισκέτα όπου θα αναγράφονται ευκρινώς τα στοιχεία σας, (β) το φυλλάδιο εκφώνησης συμπληρωμένο, και (γ) οι γραφικές παραστάσεις.

Υποδείξεις. Η εργασία είναι **ατομική**. 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 να γίνει με τον Αριθμό Μητρώου σας. Στον παρακάτω κώδικα έχει χρησιμοποιηθεί ο (φανταστικός) Αριθμός Μητρώου 2005143.

```
#include <stdlib.h>

#define LIMIT 30000

srand((long) 2005143);          /* Αρχικοποίηση γεννήτριας τυχαίων αριθμών με AM */
for (i = 0; i < n; i++)
    A[i] = 1 + rand() % LIMIT; /* n τυχαίοι αριθμοί στο διάστημα [1,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;
```