

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΕΠΛ 035: Δομές Δεδομένων και Αλγόριθμοι για Ηλεκτρολόγους Μηχανικούς και
Μηχανικούς Υπολογιστών**

Χειμερινό Εξάμηνο 2012

ΑΣΚΗΣΗ 4

**Εμπειρική Ανάλυση Χρονική Πολυπλοκότητας Αλγόριθμων
Ταξινόμησης**

Διδάσκων Καθηγητής: Παναγιώτης Ανδρέου

Ημερομηνία Υποβολής: 22/10/2012

Ημερομηνία Παράδοσης: 01/11/2012

Υπεύθυνος Εργαστηρίου: Παύλος Αντωνίου (paul.antoniou@cs.ucy.ac.cy)

ΠΕΡΙΓΡΑΦΗ

Σε αυτή την άσκηση καλείστε να υλοποιήσετε ένα πρόγραμμα το οποίο θα διεξάγει εμπειρική ανάλυση της χρονικής πολυπλοκότητας των αλγορίθμων ταξινόμησης που διδαχθήκατε στις διαλέξεις.

ΑΣΚΗΣΗ 1

Για αυτή την άσκηση πρέπει να υλοποιήσετε όλους τους αλγόριθμους ταξινόμησης που υπάρχουν στις διαλέξεις. Αυτό περιλαμβάνει τους αλγόριθμους **SelectionSort**, **InsertionSort**, **BubbleSort**, **MergeSort**, **QuickSort** και **BucketSort**. Σημειώστε ότι για κάθε αλγόριθμο, μπορείτε να χρησιμοποιήσετε σαν πρότυπο τον κώδικα που έχει παραδοθεί στις διαλέξεις.

Επίσης, για την καταγραφή του χρόνου, θα πρέπει να χρησιμοποιήσετε την βιβλιοθήκη `time.h` που διδαχθήκατε στο πρώτο εργαστήριο.

Το ζητούμενο της άσκησης είναι να δημιουργήσετε ένα πρόγραμμα που τρέχει από την γραμμή εντολής και εκτελεί τα εξής βήματα:

1. Δέχεται σαν παράμετρο εισόδου ένα ακέραιο αριθμό $n > 0$. (Στην περίπτωση που δοθεί λάθος αριθμός, το μήνυμα τυπώνει μήνυμα λάθους πριν την έξοδο του.)
2. Θα δημιουργεί ένα δυναμικό πίνακα ακεραίων (`array []`) με μέγεθος n ;
3. Θα γεμίζει τον πίνακα `array` με τρεις διαφορετικούς τρόπους:
 - a. Με τυχαίους αριθμούς, `array_random []`
 - b. Με αριθμούς από 0 ως n , `array_asc []`
Ο πίνακας `array_asc` είναι ταξινομημένος σε αύξουσα σειρά
 - c. Με αριθμούς από n ως 0, `array_desc []`
Ο πίνακας `array_desc` είναι ταξινομημένος σε φθίνουσα σειρά
4. Ακολούθως, θα καλεί κάθε αλγόριθμο ταξινόμησης να ταξινομήσει τους πιο πάνω πίνακες. Το πρόγραμμα θα μετράει τον χρόνο από τη στιγμή που καλείται κάποιος αλγόριθμος, μέχρι τον χρόνο που τερματίζει.
5. Για κάθε αποτέλεσμα του βήματος 4, θα πρέπει να τυπώνεται το n , το όνομα του αλγόριθμου, το όνομα του πίνακα και τέλος ο χρόνος ταξινόμησης.

ΑΣΚΗΣΗ 2

Χρησιμοποιώντας το πρόγραμμα που αναπτύξατε στην άσκηση 1, θα πρέπει να παράξετε μία αναφορά (με γραφικές παραστάσεις με στήλες-bar chart) η οποία θα περιλαμβάνει πειράματα για τις εξής τιμές του n .

- a) 10
- b) 50
- c) 100
- d) 10,000
- e) 100,000
- f) 1,000,000
- g) Ένα n δικής σας επιλογής, μεγαλύτερο του 1,000,000

Η αναφορά θα πρέπει να έχει την εξής δομή:

- Μία σελίδα για κάθε είδος πίνακα (τυχαίο, ταξινομημένο σε αύξουσα σειρά, ταξινομημένο σε φθίνουσα σειρά)
- Μία γραφική παράσταση για κάθε τιμή του n που θα δείχνει τον χρόνο εκτέλεσης του κάθε αλγόριθμου:
 - Στον Y-άξονα θα καταγράφετε τον χρόνο
 - Στον X-άξονα θα δείχνετε το όνομα του αλγόριθμου

ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση του προγράμματός σας, θα ληφθούν οι πιο κάτω παράμετροι:

- **Ορθότητα:** ο πρόγραμμά σας πρέπει τρέχει ορθά για οποιαδήποτε είσοδο
- **Κατανοητά σχόλια:** Γράφετε κατανοητά σχόλια που να εξηγούν την λειτουργία της κάθε δομής/συνάρτησης.
- **Αποδοτικότητα:** το πρόγραμμά σας θα πρέπει να δεσμεύει/αποδεσμεύει την μνήμη αποδοτικά.

ΟΔΗΓΙΕΣ

- Οι άσκηση σας θα πρέπει να παραδοθεί σε ηλεκτρονική μορφή μέσω email στον υπεύθυνο βοηθό σαν ένα αρχείο (.c) ή/και αρχεία (.h), που να περιλαμβάνει τον κώδικά σας.
- Το αρχείο (.c) θα ονομάζεται με τον εξής τρόπο:
erl035.ex<αριθμός άσκησης>.<ταυτότητα>.c (π.χ., erl035.ex4.123456.c)
- Στην περίπτωση χρήσης βιβλιοθήκης (.h) τότε το αρχείο θα ονομάζεται με τον εξής τρόπο:
erl035.ex<αριθμός άσκησης>.<ταυτότητα>.<όνομα βιβλιοθήκης>.h
(π.χ., erl035.ex4.123456.stack.h)
- Βεβαιωθείτε ότι τα προγράμματα σας είναι ορθά και τρέχουν.
- Τα αρχεία σας πρέπει να μεταγλωττίζονται ορθά με τον μεταγλωττιστή (compiler) gcc του εργαστηρίου UNIX. Παράδειγμα μεταγλώττισης:
gcc erl035.ex4.123456.c -o a.out
- Στην αρχή του κάθε αρχείου θα πρέπει να αναγράφεται η ταυτότητά σας σε σχόλια.
- Επίσης θα πρέπει να παραδώσετε μία αναφορά, σε έντυπη μορφή η οποία θα περιέχει τον κώδικα σας και ένα αντιπροσωπευτικό παράδειγμα εκτέλεσης του μαζί με τα σχόλια που ζητούνται πιο πάνω.
- Στην αναφορά σας θα πρέπει να αναφέρετε συγκεκριμένα ποιες συναρτήσεις δουλεύουν και ποιες όχι.
- Μη τήρηση των ημερομηνιών παράδοσης των εργασιών συνεπάγεται τις ανάλογες βαθμολογικές επιπτώσεις (μέχρι τον μηδενισμό της εργασίας).
- Οι προγραμματιστικές ασκήσεις θα ελέγχονται από ειδικό πρόγραμμα για την ανίχνευση των αντιγραφών. Οι αντιγραμμένες εργασίες θα μηδενίζονται και για τους αντιγραφείς θα εφαρμόζονται οι κανόνες τού Πανεπιστημίου. Αποφύγετε λοιπόν την αντιγραφή προγραμμάτων από άλλους συναδέλφους σας, διότι έτσι εκτίθετε και αυτούς και τον εαυτό σας στον κίνδυνο μηδενισμού και πειθαρχικής δίωξης.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!