
PROJECT: “DATA STRUCTURES 2019”

DEADLINE: ΗΜΕΡΟΜΗΝΙΑ ΕΞΕΤΑΣΗΣ ΤΟΝ ΙΟΥΝΙΟ

(Αποστολή της άσκησης με e-mail στο sioutas@ceid.upatras.gr, makri@ceid.upatras.gr, kanavos@ceid.upatras.gr, mvonitsanos@ceid.upatras.gr, asavv@ceid.upatras.gr)

PART I: “Sorting and Searching Algorithms”

- (1) Υλοποιήστε σε γλώσσα C τους αλγορίθμους ταξινόμησης **Bubble Sort**, **Insertion Sort** και **Selection sort**, σύμφωνα με τον ψευδοκώδικα που σας επεξηγήθηκε στη θεωρία (για λεπτομέρειες δείτε τις σχετικές διαφάνειες στο e-class). Χρησιμοποιώντας το ίδιο Data Set, συγκρίνατε πειραματικά τους τρεις (3) αλγορίθμους. Τι παρατηρείτε?
- (2) Υλοποιήστε σε γλώσσα C τους αλγορίθμους ταξινόμησης **Merge Sort** και **Quick Sort**, σύμφωνα με τον ψευδοκώδικα που σας επεξηγήθηκε στη θεωρία (για λεπτομέρειες δείτε τις σχετικές διαφάνειες στο e-class). Χρησιμοποιώντας το ίδιο Data Set, συγκρίνατε πειραματικά τους δύο (2) αλγορίθμους. Τι παρατηρείτε?
- (3) Υλοποιήστε σε γλώσσα C τον αλγόριθμο ταξινόμησης **Heap Sort**. Συμβουλευτείτε τον ψευδοκώδικα της σελίδας 47 του βιβλίου («Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής) που είναι διαθέσιμο στο e-class. Χρησιμοποιώντας το ίδιο Data Set, συγκρίνατε πειραματικά τους έξι (6) παραπάνω αλγορίθμους ταξινόμησης. Τι παρατηρείτε? Επαληθεύεται η θεωρία?
- (4) Υλοποιήστε σε γλώσσα C τους αλγορίθμους **Γραμμικής Αναζήτησης**, **Διαδικής Αναζήτησης** και **Αναζήτησης με Παρεμβολή**. Χρησιμοποιώντας το ίδιο Data Set, συγκρίνατε πειραματικά τους παραπάνω τρεις αλγορίθμους αναζήτησης. Τι παρατηρείτε ως προς τους χρόνους μέσης περίπτωσης? Πόσο η ΚΑΤΑΝΟΜΗ του Data Set επηρεάζει την απόδοση του κάθε αλγορίθμου?
- (5) Υλοποιήστε σε γλώσσα C τον αλγόριθμο **Διαδικής Αναζήτησης Παρεμβολής (BIS)**. Συμβουλευτείτε τον ψευδοκώδικα της σελίδας 82 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις διαφάνειες *9.searching.pdf* που είναι διαθέσιμα στο e-class. Επαληθεύστε πειραματικά τη χρονική πολυπλοκότητα που ισχύει για την μέση (expected) και χειρότερη περίπτωση (worst-case). Η βελτίωση της χειρότερης περίπτωσης επιτυγχάνεται με μία παραλλαγή του BIS. Συμβουλευτείτε τη σελίδα 85 του βιβλίου «Δομές Δεδομένων», Α.Κ. Τσακαλίδης, Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Η/Υ και Πληροφορικής καθώς και τις διαφάνειες *9.searching.pdf* που είναι διαθέσιμα στο e-class και υλοποιήστε τον αλγόριθμο της συγκεκριμένης παραλλαγής του **BIS**. Χρησιμοποιώντας το ίδιο Data Set, συγκρίνατε πειραματικά τους παραπάνω δύο αλγορίθμους. Τι παρατηρείτε ως προς τους χρόνους χειρότερης περίπτωσης?

PART II: BSTs & HASHING

- (A) Με τον κατάλληλο ορισμό δομών (structs) και συναρτήσεων (functions), να υλοποιήσετε μια εφαρμογή (να γράψετε ένα πρόγραμμα σε γλώσσα C) που θα επεξεργάζεται δεδομένα βαθμολογίου φοιτητών τμήματος Πληροφορικής. Τα δεδομένα θα πρέπει να φορτώνονται από ένα αρχείο κειμένου που περιέχει την εξής πληροφορία για κάθε φοιτητή (ανά γραμμή του

αρχείου): AM φοιτητή (π.χ. AE797989), το όνομα του φοιτητή (π.χ. «Γιώργος»), το επίθετο του φοιτητή (π.χ. «Γεωργόπουλος»), τη βαθμολογία του στο μάθημα των ΔΟΜΩΝ ΔΕΔΟΜΕΝΩΝ (π.χ. 9.0). Η εφαρμογή διαβάζει αρχικά το αρχείο και δημιουργεί ένα **Διαδικό Δένδρο Αναζήτησης** (ΔΔΑ) στο οποίο κάθε κόμβος του διατηρεί μια εγγραφή του αρχείου. Το ΔΔΑ διατάσσεται ως προς τον AM της εγγραφής του φοιτητή και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του ΔΔΑ η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Απεικόνιση του ΔΔΑ με ενδο-διατεταγμένη διάσχιση. Κάθε απεικόνιση θα πρέπει να περιέχει μια επικεφαλίδα με τους τίτλους των στοιχείων που απεικονίζονται.
2. Αναζήτηση ενός φοιτητή βάσει του AM του (θα δίνεται από το πληκτρολόγιο) και απεικόνιση της πληροφορίας για την εγγραφή του φοιτητή.
3. Τροποποίηση των καταχωρημένων στοιχείων του φοιτητή βάσει AM που θα δίνεται από το πληκτρολόγιο. Η τροποποίηση μπορεί να αφορά οποιοδήποτε πεδίο της εγγραφής του φοιτητή (εκτός του AM της).
4. Διαγραφή μιας εγγραφής φοιτητή βάσει AM που θα δίνεται από το πληκτρολόγιο.
5. Έξοδος από την εφαρμογή.

(B) Τροποποιήστε κατάλληλα τον κώδικα του (A), ώστε το αρχείο να διαβάζεται στο ΔΔΑ με βάση τη βαθμολογία του φοιτητή στο μάθημα ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ. Το ΔΔΑ διατάσσεται ως προς τη βαθμολογία της εγγραφής του φοιτητή και υλοποιείται με δυναμική διαχείριση μνήμης. Μετά την δημιουργία του ΔΔΑ η εφαρμογή εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Εύρεση των φοιτητών με την ΕΛΑΧΙΣΤΗ βαθμολογία
2. Εύρεση των φοιτητών με τη ΜΕΓΙΣΤΗ βαθμολογία

(Γ) Υλοποιήστε το (A) κάνοντας χρήση HASHING με αλυσίδες, αντί ΔΔΑ. Η συνάρτηση κατακερματισμού θα υπολογίζεται ως το υπόλοιπο (modulo) της διαίρεσης του αθροίσματος των κωδικών ASCII των επιμέρους χαρακτήρων που απαρτίζουν τον AM με ένα περιττό αριθμό m που συμβολίζει το πλήθος των κάδων (buckets). Π.χ. για AM="AE797989" και m=11, ισχύει:

$$\text{Hash}(\text{AE797989}) = [\text{ASCII}('A') + \text{ASCII}('E') + \text{ASCII}('7') + \text{ASCII}('9') + \text{ASCII}('7') + \text{ASCII}('9') + \text{ASCII}('8') + \text{ASCII}('9')] \bmod m.$$

Το πρόγραμμα θα εμφανίζει ένα μενού με τις ακόλουθες επιλογές:

1. Αναζήτηση ενός φοιτητή βάσει του AM του (θα δίνεται από το πληκτρολόγιο) και απεικόνιση της πληροφορίας για την εγγραφή του φοιτητή.
2. Τροποποίηση των στοιχείων εγγραφής φοιτητή βάσει AM που θα δίνεται από το πληκτρολόγιο. Η τροποποίηση μπορεί να αφορά οποιοδήποτε πεδίο της εγγραφής του φοιτητή (εκτός του AM της).
3. Διαγραφή μιας εγγραφής φοιτητή από τον πίνακα κατακερματισμού βάσει AM που θα δίνεται από το πληκτρολόγιο.
4. Έξοδος από την εφαρμογή.

Ενοποιήστε τα (A), (B) και (Γ) σε ένα πρόγραμμα στο οποίο ο χρήστης θα ερωτάται αν θέλει τη φόρτωση του αρχείου σε ένα ΔΔΑ ή σε μία δομή Hashing με αλυσίδες και στην περίπτωση που ο χρήστης επιλέξει το πρώτο να μπορεί εν συνεχεία να επιλέξει αν η φόρτωση στο ΔΔΑ θα γίνει με βάση τον AM του φοιτητή ή με βάση τη βαθμολογία του εμφανίζοντας τα αντίστοιχα μενού λειτουργιών.

ΑΡΙΘΜΟΣ ΦΟΙΤΗΤΩΝ ΑΝΑ ΟΜΑΔΑ <=3

ΠΟΣΟΣΟΣΤΟ ΕΠΙ ΤΟΥ ΣΥΝΟΛΙΚΟΥ ΒΑΘΜΟΥ: 30%

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!