

Χειμερινό Εξάμηνο - Ακαδημαϊκό Έτος 2016-17

Κάθε φοιτητής πριν ξεκινήσει να καταγράφει τις απαντήσεις του, πρέπει να γράψει στην κόλλα του τα στοιχεία του και στην πάνω δεξιά γωνία της κόλλας τον αριθμό 1 (που υποδηλώνει πως αυτή είναι η πρώτη κόλλα που χρησιμοποιείται από το φοιτητή). Το διαγώνισμα διαρκεί 2 ώρες. Δεν επιτρέπεται η χρήση του βιβλίου ή άλλων σημειώσεων κατά τη διάρκεια του διαγωνίσματος. Επιτρέπεται ωστόσο κάθε φοιτητής να έχει μαζί του μια σελίδα (όχι φύλλο) Α4 με σημειώσεις.

Κάθε φοιτητής πρέπει να διαβάσει καλά τις εκφωνήσεις και να βεβαιωθεί πως καταλαβαίνει τι ζητάει το κάθε θέμα.

Θα πρέπει να γίνει καλή χρήση του χρόνου.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ**ΟΝΟΜΑΤΕΠΩΝΥΜΟ:****ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ:**

TMHMA:

ΕΤΟΣ ΣΠΟΥΔΩΝ:

Πίνακας Αυτοαξιολόγησης

[illegible]

[8%]

$\sqrt{n^3} \log(\sqrt{n^3})$	
$n^{1.5} + 2 n \log(n^{4n})$	
$n + 2 n \log(n^{4n})$	

[4%]

$$f_2(n) = 10 \sqrt{n} \log^2 n,$$

$$f_4(n) = n^{\sqrt{n}} + n^2 \sqrt{n} + 3^{350000} n.$$

γ. Έστω $T(n)$ το συνολικό πλήθος των φορών που εκτελείται η εντολή $x = x + 1$ από τη ρουτίνα `Tricky()`. Υπολογίστε το $T(n)$ και μελετήστε την τάξη του. [8%]

```
procedure Tricky(int n) {  
    for j = 1 to n do  
        for k = 1 to  $n^2 - j$  do  
             $x = x + 1$ ;  
}
```

- δ. Σχεδιάστε το δένδρο αναδρομής για την αναδρομική σχέση $T(n) = 3T(n/3) + 7$ (δεδομένου πως ισχύει ότι $T(1) = 1$). Ποια είναι η ασυμπτωτική τάξη της συνάρτησης; [5%]

Άσκηση 2 [10%]

Έστω ότι μια βιβλιοθήκη σας παρέχει πρόσβαση σε στοίβες και ουρές ακεραίων. Η βιβλιοθήκη σας επιτρέπει να ορίσετε μια στοίβα (ή μια ουρά) και να καλέσετε τις 5 βασικές λειτουργίες σε αυτή. Για παράδειγμα, ο ορισμός μιας στοίβας (ή μιας ουράς) γίνεται γράφοντας: `Stack S1;` (αντίστοιχα, `Queue Q1;`). Για τη στοίβα υποστηρίζονται οι εξής λειτουργίες: (1) `void MakeEmptyStack(stack S);` (2) `int IsEmptyStack(stack S);` (3) `int Top(Stack S);` (4) `int Pop(Stack S);` (5) `void Push(Stack S, int x);` Αντίστοιχα, για την ουρά υποστηρίζονται οι λειτουργίες: (1) `void MakeEmptyQueue(Queue Q);` (2) `int IsEmptyQueue(Queue Q);` (3) `int Front(Queue Q);` (4) `int Dequeue(Queue Q);` (5) `void Enqueue(Queue Q, int x);`

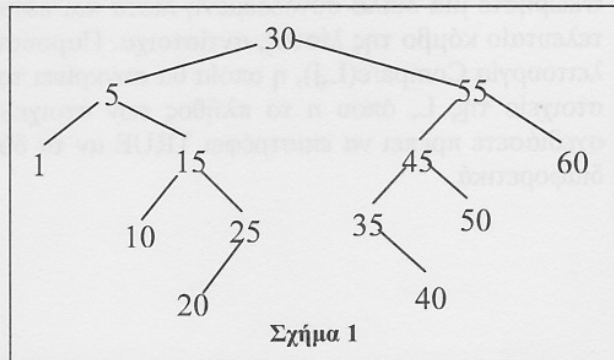
Παρουσιάστε ψευδοκώδικα για μια συνάρτηση η οποία θα υλοποιεί τη λειτουργία `int SearchStack(Stack S, int num)`, η οποία αναζητά στοιχείο με αριθμό `num` στη στοίβα. Η `SearchStack()` επιστρέφει 1 αν βρεθεί στοιχείο με αριθμό `num` και 0 διαφορετικά. Η εκτέλεση της `SearchStack()` δεν θα πρέπει να επηρεάζει τη μορφή της στοίβας (δηλαδή η στοίβα θα πρέπει να περιέχει τα ίδια στοιχεία και με την ίδια σειρά πριν και μετά την εκτέλεση της `SearchStack()`).

Άσκηση 3 [15%]

- α. Δίνεται ένα δυαδικό δένδρου με 5 κόμβους και κλειδιά A, B, C, D, E. Είναι γνωστό πως η ενδοδιατεταγμένη διάσχιση των κόμβων του δένδρου δίνει τα κλειδιά τους με την εξής διάταξη: B, E, D, A, C, ενώ η προδιατεταγμένη διάσχιση δίνει τη διάταξη A, B, D, E, C στα κλειδιά των κόμβων. Ζωγραφίστε το δένδρο. Για κάθε κόμβο εξηγήστε βάσει ποιου σκεπτικού αποφασίσατε να τον ζωγραφίσετε στη θέση που βρίσκεται στο δένδρο που σχεδιάσατε. [10%]

Σημείωση: Η παρουσίαση ενός δένδρου χωρίς δικαιολόγηση θα βαθμολογείται με 0.

- β. Ποιες είναι οι ακολουθίες κλειδιών που δίνουν οι διασχίσεις ενδοδιατεταγμένη, μεταδιατεταγμένη και προδιατεταγμένη στο δένδρο του Σχήματος 1; [5%]



Άσκηση 4 [15%]

Θεωρήστε μια διπλά-συνδεδεμένη λίστα και έστω S και F δείκτες στον πρώτο και στον τελευταίο κόμβο της λίστας, αντίστοιχα. Παρουσιάστε ψευδοκώδικα που θα υλοποιεί τη λειτουργία $\text{Compare}(L, j)$, η οποία θα συγκρίνει το j -οστό στοιχείο της L με το $(n-j)$ -οστό στοιχείο της L , όπου n το πλήθος των στοιχείων της λίστας. Ο αλγόριθμος που θα σχεδιάσετε πρέπει να επιστρέφει TRUE αν τα δύο αυτά στοιχεία είναι ίδια και FALSE διαφορετικά.

Άσκηση 5 [10%]

Παρουσιάστε αλγόριθμο, ο οποίος θα παίρνει ως όρισμα ένα δείκτη στη ρίζα ενός δυαδικού απλά-συνδεδεμένου δένδρου με κόμβους που περιέχουν ακέραια κλειδιά και **θα επιστρέφει** το μέγιστο κλειδί μεταξύ των κλειδιών των κόμβων του δένδρου. Ο αλγόριθμός σας δεν επιτρέπεται να χρησιμοποιεί static ή global μεταβλητές.

Άσκηση 6 [10%]

Σας ζητείται να γράψετε μια συνάρτηση Boolean ChildrenNodes(struct treenode *R, int k), η οποία θα παίρνει ως παράμετρο ένα δείκτη στη ρίζα ενός δυαδικού δένδρου που αναπαριστά ένα διατεταγμένο δένδρο (το οποίο δεν είναι απαραίτητα δυαδικό) και έναν ακέραιο k. Η συνάρτηση θα πρέπει να εξετάζει αν στο δένδρο υπάρχει κάποιος κόμβος που έχει τουλάχιστον k παιδιά. Αν υπάρχει τέτοιος κόμβος, η συνάρτηση θα πρέπει να επιστρέφει TRUE, ενώ στην αντίθετη περίπτωση θα πρέπει να επιστρέφει FALSE.

Άσκηση 7 [15%]

Ένα δυαδικό μεταδιατεταγμένο (postorder) ταξινομημένο δένδρο είναι ένα δυαδικό δένδρο για το οποίο η μεταδιατεταγμένη διάσχιση δίνει τα κλειδιά που είναι αποθηκευμένα στους κόμβους του δένδρου κατά αύξουσα διάταξη. Παρουσιάστε ψευδοκώδικα για έναν αλγόριθμο εύρεσης του κόμβου με κλειδί k σε δυαδικό μεταδιατεταγμένο ταξινομημένο δένδρο. Ο αλγόριθμός σας θα πρέπει να έχει χρονική πολυπλοκότητα $O(h)$.