

Χειμερινό Εξάμηνο - Ακαδημαϊκό Έτος 2014-15

ΔΙΑΓΩΝΙΣΜΑ ΠΡΟΟΔΟΥ - Νοέμβριος 2014

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]

Άσκηση 1 [25%]

α. Παρουσιάστε αυστηρά ασυμπτωτικά άνω φράγματα για τις παρακάτω συναρτήσεις.
[8%]

Συνάρτηση	Αυστηρό Ασυμπτωτικό Άνω Φράγμα
$5 \log n^{10} + 5 \sqrt{n}$	$O(\quad)$
$4n^3 \log n - n^2 \sqrt{n}$	$O(\quad)$
$n^{5/2} - 5n^4 \sqrt{n} \log^2 n$	$O(\quad)$
$8n \log n - 4n + 3 \sqrt{n} \log n$	$O(\quad)$

Κάθε εγγραφή του πίνακα συνεισφέρει 2 μονάδες στο βαθμός σας. ΛΑΘΟΣ ΕΓΓΡΑΦΕΣ μετράνε ως -1 μονάδα (δηλαδή υπάρχει αρνητική βαθμολογία).

β. Ταξινομήστε τις παρακάτω τέσσερις συναρτήσεις, σε αύξουσα διάταξη, βάσει της ασυμπτωτικής τους συμπεριφοράς.
[4%]

$$f_1(n) = 5n \log n + 3n, \quad f_2(n) = 6n^2 - 5n + 4^{10000000}, \quad f_3(n) = 2^{500000} n + n\sqrt{n},$$

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

29 Νοεμβρίου 2014

- γ. Για κάθε μια από τις αναδρομικές σχέσεις που παρουσιάζονται στη Στήλη 1 του παρακάτω πίνακα, συμπληρώστε (στην αντίστοιχη θέση της Στήλης 2) αυστηρό ασυμπτωτικό άνω φράγμα για την $T(n)$. [8%]

Αναδρομική Σχέση	Αυστηρότερο Ασυμπτωτικό Άνω Φράγμα
$T(n) = T(n/4) + c$	
$T(n) = T(n-2) + c$	
$T(n) = 4T(n/4) + cn$	
$T(n) = 3T(n/3) + c$	

Κάθε εγγραφή του πίνακα συνεισφέρει 2 μονάδες στο βαθμός σας. ΛΑΘΟΣ ΕΓΓΡΑΦΕΣ μετράνε ως -1 μονάδα (δηλαδή υπάρχει αρνητική βαθμολογία).

- δ. Σχεδιάστε το δένδρο αναδρομής για την αναδρομική σχέση $T(n) = 4T(n/4) + cn$ (δεδομένου πως ισχύει ότι $T(1) = 1$). [5%]

Ασκηση 2 [10%]

Θεωρήστε ένα δυαδικό δέντρο το οποίο είναι γεμάτο, έχει ύψος τρία και περιέχει τα εξής 11 κλειδιά: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, με κάθε υποδένδρο της ρίζας να περιέχει πέντε κλειδιά. Όταν εφαρμόζεται η προδιατεταγμένη διάσχιση πάνω στο δένδρο, με τη Visit() να τυπώνει απλά το κλειδί του εκάστοτε κόμβου, τα κλειδιά των κόμβων τυπώνονται σε **φθίνουσα** διάταξη. Σχεδιάστε ένα δένδρο που να πληροί τις προδιαγραφές της εκφώνησης.

Προσοχή: Δεν υπάρχει μόνο ένα τέτοιο δένδρο. Ζητείται απλά ένα από τα δένδρα που πληρούν τις προδιαγραφές της εκφώνησης.

Άσκηση 3 [10%]

Δίδεται ένα δυαδικό δένδρο με 5 κλειδιά. Όταν εφαρμόζεται η προδιατεταγμένη διάσχιση πάνω στο δένδρο, με τη Visit() να τυπώνει το κλειδί του εκάστοτε κόμβου, τα κλειδιά των κόμβων τυπώνονται με την ακόλουθη διάταξη: 12, 22, 21, 16, 24. Όταν εφαρμόζεται η ενδοδιατεταγμένη διάσχιση πάνω στο δένδρο με την ίδια Visit(), τα κλειδιά των κόμβων τυπώνονται με τη ακόλουθη διάταξη 12, 16, 21, 22, 24.

Ζωγραφίστε το δένδρο.

Άσκηση 4 [10%]

Δίδονται οι ακόλουθες λειτουργίες σε πολυσύνολα (δηλαδή σε σύνολα που μπορεί να περιέχουν περισσότερα από ένα αντίγραφα του ίδιου κλειδιού):

- $\text{Search}(S, K)$: Αναζητά το κλειδί K στο πολυσύνολο S .
- $\text{Insert}(S, K)$: Εισάγει το κλειδί K στο πολυσύνολο S (αν το K υπάρχει ήδη στο S , θα εισαχθεί ένα ακόμη αντίγραφο του K στο S).
- $\text{Delete}(S, K)$: Διαγράφει το κλειδί K από το S . Αν υπάρχουν περισσότερα από ένα αντίγραφα του K στο S , αρκεί να διαγραφεί ένα οποιοδήποτε από αυτά.
- $\text{Split}(S, K)$: Διαχωρίζει το S σε δύο νέα πολυσύνολα, S_1 και S_2 , τέτοια ώστε το S_1 περιέχει εκείνα τα στοιχεία του S που είναι μικρότερα ή ίσα του K , ενώ το S_2 περιέχει εκείνα τα στοιχεία του S που είναι μεγαλύτερα του K . Η $\text{Split}()$ μπορεί να αλλοιώνει, κατά την εκτέλεση της, τα S_1 και S_2 .
- $\text{Merge}(S_1, S_2)$: Δημιουργεί ένα νέο πολυσύνολο S που προκύπτει από τη συνένωση των S_1 και S_2 . Επομένως, το S περιέχει τόσα στοιχεία, όσα το άθροισμα των στοιχείων των S_1 και S_2 . Σημειώνεται επίσης ότι αν το S_1 περιέχει 3 αντίγραφα ενός κλειδιού K και το S_2 περιέχει επίσης το K σε δύο αντίγραφα, τότε το S θα περιέχει 5 αντίγραφα του K . Η $\text{Merge}()$ επιτρέπεται να αλλοιώνει, κατά την εκτέλεση της, τα S_1 και S_2 .

Συμπληρώστε στις αντίστοιχες θέσεις του πίνακα, την τάξη της χρονικής πολυπλοκότητας (χρησιμοποιώντας το συμβολισμό Θ) που θα είχε κάθε μια από τις παραπάνω λειτουργίες στην καλύτερη υλοποίηση που μπορείτε να σκεφτείτε χρησιμοποιώντας κάθε μια από τις δομές που αναγράφονται στις γραμμές του πίνακα. Στην περίπτωση της μη-ταξινομημένης, απλά-συνδεδεμένης λίστας, θεωρήστε ότι η δομή υλοποιείται με δύο δείκτες, έναν που δείχνει στο πρώτο στοιχείο της λίστας και έναν που δείχνει στο τελευταίο στοιχείο της.

Λειτουργίες Δομή Δεδ.	Search	Insert	Delete	Split	Merge
μη-ταξινομημένη, στατική λίστα					
ταξινομημένη, στατική λίστα					
μη-ταξινομημένη, απλά-συνδεδεμένη λίστα					
ταξινομημένη, απλά- συνδεδεμένη λίστα					
μη-ταξινομημένο δυναμικό δένδρο					

Κάθε εγγραφή του πίνακα συνεισφέρει 0.4 μονάδες στο βαθμός σας. ΛΑΘΟΣ ΕΓΓΡΑΦΕΣ μετράνε ως -0.2 μονάδες (δηλαδή υπάρχει αρνητική βαθμολογία).

Άσκηση 5 [15%]

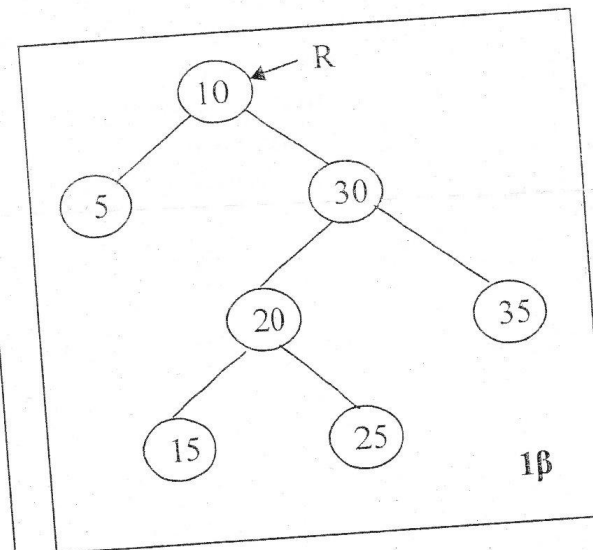
Δίδονται δύο σύνολα S_1 και S_2 που υλοποιούνται χρησιμοποιώντας ταξινομημένες, απλά-συνδεδεμένες λίστες. Ζητείται ψευδοκώδικας που θα υλοποιεί τη λειτουργία `SetMinus()` επί των συνόλων S_1 και S_2 . Η `SetMinus()` δέχεται ως παραμέτρους δύο δείκτες, έναν στο πρώτο στοιχείο της λίστας που υλοποιεί το S_1 και έναν στο πρώτο στοιχείο της λίστας που υλοποιεί το S_2 και επιστρέφει έναν δείκτη στο πρώτο στοιχείο μιας νέας λίστας η οποία περιέχει εκείνα τα στοιχεία του S_1 που δεν ανήκουν στο S_2 (δηλαδή που περιέχει τα στοιχεία του συνόλου $S_1 - S_2$). Οι λίστες που υλοποιούν τα S_1 και S_2 θα πρέπει να μείνουν ανέπαφες κατά την εκτέλεση του αλγορίθμου.

Άσκηση 6 [20%]

- α. Ποια είναι η τιμή της μεταβλητής temp του αλγορίθμου του Σχήματος 1α, αμέσως πριν τον τερματισμό του αλγορίθμου, όταν αυτός εφαρμόζεται με όρισμα ένα δείκτη στη ρίζα του δένδρου του Σχήματος 1β; [8%]

```
void WhatDoIPrint(NODE *R) {  
    static int temp = 0;  
  
    If (R == NULL) return 0;  
  
    temp = temp + 2 * R->key;  
  
    WhatDoIPrint(R->LC)  
    WhatDoIPrint(R->RC)  
  
    temp++;  
}
```

1α



Σχήμα 1

ΤΙΜΗ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ temp:

- β. Παρουσιάστε αλγόριθμο, ο οποίος θα παίρνει ως όρισμα ένα δείκτη στη ρίζα ενός δυαδικού απλά-συνδεδεμένου δένδρου και **θα επιστρέφει** το άθροισμα των κλειδιών των κόμβων του δένδρου. Ο αλγόριθμός σας δεν επιτρέπεται να χρησιμοποιεί static ή global μεταβλητές. [12%]

Άσκηση 7 [10%]

Γράψτε μια διαδικασία `RemoveChild(pointer r)` η οποία θα παίρνει ως όρισμα ένα δείκτη `R` στη ρίζα ενός δυαδικού δένδρου που αναπαριστά (προσομοιώνει) ένα (όχι απαραίτητα δυαδικό) διατεταγμένο δένδρο. Θεωρήστε ότι κάθε κόμβος του δένδρου αποθηκεύει εκτός από τους απαραίτητους δείκτες (`lc` και `rs`) και έναν ακέραιο `num`. Η συνάρτηση θα πρέπει να διασχίζει όλους τους κόμβους του διατεταγμένου δένδρου και για κάθε κόμβο `v` θα πρέπει να κάνει τα εξής: Αν το δεξιότερο παιδί του `v` στο διατεταγμένο δένδρο είναι φύλλο, το παιδί αυτό θα πρέπει να διαγράφεται.