Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών 19 Νοεμβρίου 2013

ΗΥ240: Δομές Δεδομένων

Διδάσκουσα: Παναγιώτα Φατούρου

Χειμερινό Εξάμηνο - Ακαδημαϊκό Έτος 2013-14

ΔΙΑΓΩΝΙΣΜΑ ΠΡΟΟΔΟΥ - Νοέμβριος 2013

Κάθε φοιτητής πριν ξεκινήσει να απαντάει, 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). Τα θέματα βαθμολογούνται με 110 μονάδες. Το άριστα είναι το 100 και επομένως οι φοιτητές μπορούν να επιλέξουν τα θέματα στα οποία προτιμούν να δουλέψουν για να πάρουν άριστα. Είναι πολύ θεμιτό, αν κάποιος φοιτητής το θελήσει, να δουλέψει σε όλα τα θέματα (γραπτά που θα βαθμολογηθούν με βαθμό μεγαλύτερο του 100 θα εξασφαλίσουν κάποιο bonus στις ασκήσεις). Το διαγώνισμα διαρκεί 1 ώρα και 45 λεπτά. Δεν επιτρέπεται η χρήση του βιβλίου ή άλλων σημειώσεων κατά τη διάρκεια του διαγωνίσματος. Επιτρέπετε 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

Θέμα 1 (25 Μονάδες, 30 Λεπτά)

α. Έστω T(n) το συνολικό πλήθος των φορών που εκτελείται η εντολή x= x+1 στον παρακάτω αλγόριθμο. Υπολογίστε το T(n). Τι τάξης είναι; [10Μ, 10Λ]

```
Procedure HowManyPrimitiveInstructions (integer n) {
    for (i=0; i < n; i = i + 1)  {
        for (j = 2*i; j < n; j++)
            x = x+1;
    }
```

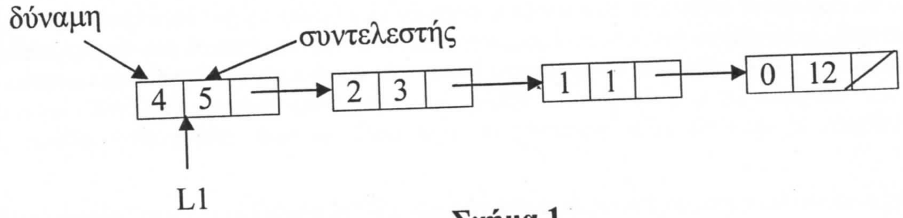
β. Για κάθε μια από τις αναδρομικές σχέσεις που παρουσιάζονται στη Στήλη 2 του παρακάτω πίνακα, συμπληρώστε (στην αντίστοιχη θέση της Στήλης 3) ποια νομίζετε πως είναι η τάξη της T(n). Συμπληρώστε επίσης πόσες αναδρομικές κλήσεις (δηλαδή πόσα υπο-προβλήματα του αρχικού προβλήματος) θα δημιουργηθούν και πόσο θα κοστίσει η εκτέλεση κάθε αναδρομικής κλήσης. [15Μ, 20Λ]

	Τάξη Πολυπλοκότητας T(n)	**Συνολικό πλήθος αναδρομικών κλήσεων**	**Κόστος κάθε αναδρομικής κλήσης**
T(n) = T(n/2) + c			
T(n) = 2 T(n/2) + c			
T(n) = 2T(n/2) + cn			
T(n) = T(n-1) + c			
T(n) = T(n-1) + cn			

Πίνακας 1

Θέμα 2 (45 Μονάδες, 40 Λεπτά)

α. Υποθέστε πως θέλουμε να χρησιμοποιήσουμε τις δομές δεδομένων που έχουμε διδαχθεί για να αναπαραστήσουμε πολυώνυμα. Για παράδειγμα, θεωρήστε το πολυώνυμο $p_1(x) = 5x^4+3x^2+x+12$. Το πολυώνυμο θα μπορούσε να αναπαρασταθεί με μια λίστα ή με ένα δυαδικό δένδρο που θα περιείχε τόσα στοιχεία όσα και οι όροι του πολυωνύμου (παρεμπιπτόντως, οι όροι του $p_1(x)$ είναι τέσσερις, οι $5x^4$, $3x^2$, x, 12, αντίστοιχα). Στο Σχήμα 1 παρουσιάζεται αναπαράσταση του πολυωνύμου $p_1(x)$ χρησιμοποιώντας μια απλά-συνδεδεμένη, ταξινομημένη (ως προς τη δύναμη στην οποία είναι υψωμένο το x σε κάθε όρο) λίστα. Κάθε στοιχείο της λίστας περιέχει τη δύναμη στην οποία υψώνεται η μεταβλητή x και τον αντίστοιχο συντελεστή. [30Μ, 30Λ]



Σχήμα 1

Σε κάθε γραμμή της πρώτης στήλης του Πίνακα 2, δίνεται μια δομή δεδομένων από αυτές που έχουμε διδαχθεί, η οποία μπορεί να χρησιμοποιηθεί για την υλοποίηση πολυωνύμων. Συγκεκριμένα, κάθε όρος του πολυώνυμου θα είναι αποθηκευμένος σε έναν κόμβο μιας τέτοιας δομής.

Επίσης, σε κάθε στήλη της πρώτης γραμμής του Πίνακα 2, δίνεται μια λειτουργία που θα θέλαμε να υλοποιηθεί πάνω σε πολυώνυμα. Κάθε τέτοια λειτουργία παίρνει ως όρισματα δύο δείκτες σε δύο δομές που υλοποιούν δύο πολυώνυμα και εφαρμόζει την αναγραφόμενη λειτουργία στα πολυώνυμα αυτά.

Συμπληρώστε στις αντίστοιχες θέσεις του πίνακα, την ασυμπτωτική χρονική πολυπλοκότητα (χρησιμοποιώντας το συμβολισμό Θ) που θα είχε κάθε λειτουργία στην καλύτερη υλοποίηση που μπορείτε να σκεφτείτε δεδομένου ότι το πολυώνυμο υλοποιείται με κάθε μια από τις δομές που αναγράφονται στις γραμμές του πίνακα.

Για κάθε δομή και κάθε λειτουργία, δικαιολογήστε σύντομα (σε 2-3 γραμμές), την πολυπλοκότητα που παρουσιάσατε στον πίνακα για τη δομή και τη λειτουργία αυτή.

Λειτουργίες / Δομή Δεδ.	Πρόσθεση Πολυωνύμων	Πολλαπλασιασμός Πολυωνύμων	Πολυώνυμο με μεγαλύτερο βαθμό	Ισότητα
μη-ταξινομημένες απλά συνδεδεμένες λίστες				
ταξινομημένες απλά συνδεδεμένες λίστες				
δυαδικά δένδρα				

Πίνακας 2

β. Θεωρήστε ένα πολυώνυμο το οποίο έχει υλοποιηθεί με μια απλά-συνδεδεμένη ταξινομημένη λίστα όπως φαίνεται στο Σχήμα 1. Παρουσιάστε ψευδοκώδικα για μια ρουτίνα η οποία θα παίρνει ως όρισμα έναν ακέραιο k (που αναπαριστά μια δύναμη του x) και έναν δείκτη στο πρώτο στοιχείο της λίστας που αναπαριστά το πολυώνυμο. Η ρουτίνα θα πρέπει να τυπώνει τον όρο του πολυωνύμου που αντιστοιχεί στην αμέσως μεγαλύτερη δύναμη από την k. [15Μ, 10Λ]

Σημείωση: Το k μπορεί να είναι δύναμη που αντιστοιχεί σε υπαρκτό όρο του πολυωνύμου ή όχι. το πολυώνυμο μπορεί να έχει οποιοδήποτε πλήθος όρων (ακόμη και 0).

Θέμα 3 (40 μονάδες, 30 λεπτά)

α. Έστω ένα (όχι απαραίτητα δυαδικό) διατεταγμένο δένδρο που έχει υλοποιηθεί με ένα δυαδικό δένδρο. Να παρουσιαστεί ψευδοκώδικας για έναν αλγόριθμο, ο οποίος θα παίρνει ως όρισμα έναν δείκτη r στη ρίζα του δυαδικού δένδρου και έναν δείκτη p σε έναν κόμβο v του δένδρου και επιστρέφει δείκτη στον κόμβο που αντιστοιχεί στο γονικό κόμβο του v στο διατεταγμένο δένδρο. [20Μ, 15Λ]

β. Δίνονται δύο σύνολα S_1 και S_2 από ακέραιους αριθμούς. Θεωρήστε πως τα στοιχεία του S_1 είναι αποθηκευμένα σε μια απλά συνδεδεμένη λίστα (ως προς το πεδίο data) ενώ τα στοιχεία του S_2 έχουν αποθηκευτεί σε ένα δυαδικό δένδρο, έτσι ώστε η μεταδιατεγμένη διάσχιση να επισκέπτεται τα στοιχεία του S_2 σε αύξουσα διάταξη. Ζητείται ψευδοκώδικας για μια συνάρτηση που θα ονομάζεται MergeListTree(). Η MergeListTree() δέχεται ως παραμέτρους δύο δείκτες, έναν στο πρώτο στοιχείο της λίστας που υλοποιεί το S_1 και έναν στη ρίζα του δένδρου που υλοποιεί το S_2 και επιστρέφει ένα δείκτη στο πρώτο στοιχείο μιας νέας ταξινομημένης λίστας η οποία θα περιέχει τα στοιχεία και των δύο συνόλων S_1 και S_2. Ο αλγόριθμός σας θα πρέπει να εκτελείται σε χρόνο $O(n_1+n_2)$, όπου n_1 και n_2 είναι το πλήθος των στοιχείων των S1 και S2, αντίστοιχα. [20Μ, 20Λ]