Η/Υ 432: Δομές Δεδομένων

Χειμερινό Εξάμηνο Ακαδημαϊκού Έτους 2007-2008

Παναγιώτα Φατούρου – Ευριπίδης Μάρκου

3ο Σετ Ασκήσεων

Ημερομηνία Παράδοσης: Οι ασκήσεις 1 και 2 για Δευτέρα, 10/12, ώρα 17:00-19:00 **(έχουν πάρει παράταση για Τρίτη, 11/12, ώρα 12:00 στο γραφείο της διδάσκουσας)**. Οι ασκήσεις 3 και 4 για Παρασκευή, 14/12, ώρα 12:00-14:00 (**έχουν πάρει παράταση για Δευτέρα, 17/12, ώρα 12:00 στο γραφείο Α25 των βοηθών)**.

Τρόπος Παράδοσης: Οι ασκήσεις παραδίδονται στους βοηθούς του μαθήματος το αργότερο μέχρι την ημερομηνία παράδοσης. Το γραφείο των βοηθών του μαθήματος είναι το Α25.

Άσκηση 1

α. Παρουσιάστε μια αναδρομική συνάρτηση, η οποία δοθέντος ενός δείκτη σε κάποιο κόμβο v ενός δυαδικού δένδρου Τ, θα επιστρέφει το ύψος του v.

β. Παρουσιάστε αλγόριθμο (ψευδοκώδικα), ο οποίος δοθείσης της ρίζας ενός δυαδικού δένδρου Τ, παράγει έναν κλώνο Τ' (ακριβές αντίγραφο) του Τ. Υποθέστε ότι κάθε κόμβος του Τ είναι της μορφής struct node {int x; struct node *lc; struct node *rc;}.

Υπόδειξη: Διασχίστε όλους τους κόμβους του Τ (με επιλογή της κατάλληλης αναδρομικής διάσχισης). Η Visit() δημιουργεί στο νέο δένδρο τα παιδιά του τρέχοντος κόμβου. Η διάσχιση πρέπει να έχει ως παράμετρο ένα δείκτη στον τρέχοντα κόμβο του Τ και έναν στον αντίστοιχο κόμβο του Τ'.

γ. Σας ζητείται να παρουσιάσετε δύο συναρτήσεις. Η πρώτη συνάρτηση θα παίρνει ως όρισμα έναν δείκτη σε κάποιον κόμβο **ενός δυαδικού δένδρου που αναπαριστά (προσομοιώνει) ένα (όχι απαραίτητα δυαδικό) διατεταγμένο δένδρο** και θα επιστρέφει το βαθμό του κόμβου. Η δεύτερη θα παίρνει ως όρισμα έναν δείκτη στη ρίζα ενός τέτοιου δένδρου και θα επιστρέφει το βαθμό του δένδρου.

Υπόδειξη: Τα δένδρα δεν είναι ταξινομημένα. Είναι χρήσιμο (παρότι δεν ζητείται) να κάνετε trace την εκτέλεση των αλγορίθμων που θα σχεδιάσετε. Για κάποιους από τους παραπάνω αλγορίθμους μπορεί να χρειαστείτε static ή global μεταβλητές. Τέτοιες μεταβλητές επιτρέπεται να χρησιμοποιήσετε.

Άσκηση 2

α. Έστω η ακολουθία Σ των γραμμάτων που απαρτίζουν το ονοματεπώνυμό σας. Για παράδειγμα, η ακολουθία αυτή για τη διδάσκουσα του μαθήματος θα ήταν Σ = {Π,Α,Ν,Α,Γ,Ι,Ω,Τ,Α,Φ,Α,Τ,Ο,Υ,Ρ,Ο,Υ}. Παρουσιάστε ένα πλήρες δυαδικό δένδρο και τοποθετήστε με τρεις τρόπους στους κόμβους του τα γράμματα της ακολουθίας του ονόματός σας, έτσι ώστε η ακολουθία Σ να προκύπτει κατά:

(i) τη μετα-διατεταγμένη διάσχιση,

(ii) την προ-διατεταγμένη διάσχιση,

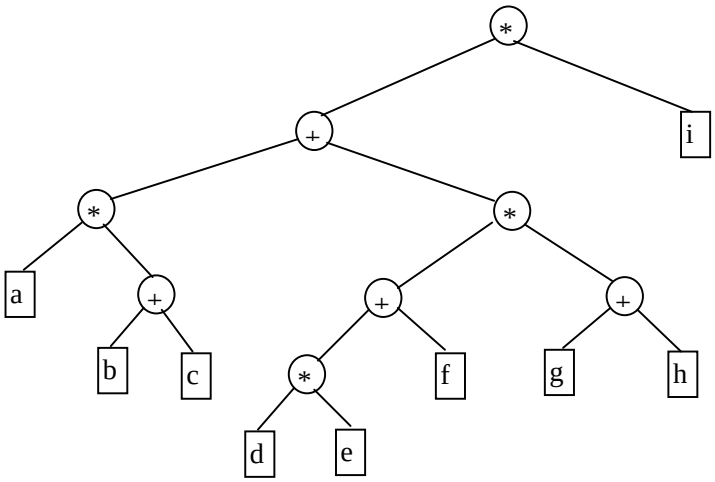
(iii) την ένδο-διατεταγμένη διάσχιση.

β. Οι αριθμητικές εκφράσεις αποτελούνται από τελεστές και μεταβλητές ή τελεστέους. Στην άσκηση αυτή εστιάζουμε σε τελεστές που είναι δυαδικοί (δηλαδή εφαρμόζονται σε δύο μεταβλητές). Ενδοθεματική (infix) ονομάζεται η μορφή παράστασης αριθμητικών εκφράσεων

στην οποία ο τελεστής εμφανίζεται μεταξύ των δύο μεταβλητών. Στην μεταθεματική μορφή παράστασης κάθε τελεστής εμφανίζεται μετά το ζεύγος των αντίστοιχων μεταβλητών, ενώ στην προθεματική ο τελεστής εμφανίζεται πριν το ζεύγος των δύο μεταβλητών.

Για παράδειγμα, η αριθμητική έκφραση (((a*(b+c))+(((d*e)+f)*(g+h)))*i) είναι σε ενδοθεματική μορφή. Η μεταθεματική ισοδύναμη έκφραση αυτής είναι abc+*de*f+gh+*+i*, ενώ η προθεματική είναι *+*a+bc*+*def+ghi.

Μια εφαρμογή των δυαδικών δένδρων αφορά τη δυνατότητα υπολογισμού αριθμητικών εκφράσεων. Για παράδειγμα, η παραπάνω αριθμητική έκφραση αντιστοιχεί στο ακόλουθο δένδρο υπολογισμού:



Παρουσιάστε αλγόριθμο (ψευδοκώδικα) που δοθείσης μιας αριθμητικής έκφρασης σε προθεματική μορφή θα δημιουργεί το ισοδύναμο δένδρο υπολογισμού. Ο αλγόριθμός σας μπορεί να είναι αναδρομικός ή μπορεί να χρησιμοποιεί βοηθητικές στοίβες (ή ουρές).

Υπόδειξη για την αναδρομική έκδοση: Η MakeTree() που χτίζει το δένδρο, θα παίρνει σαν όρισμα έναν δείκτη στον τρέχοντα κόμβο του δένδρου υπό κατασκευή. Ο δείκτης αυτός είναι NULL, όταν καλείται η MakeTree() για πρώτη φορά. Η MakeTree() λειτουργεί ως εξής:

Για δύο φορές εκτέλεσε τα εξής: /* μια για το αριστερό και μια για το δεξιό υποδένδρο */

Διάβασε χαρακτήρα;

Αν η ακολουθία δεν έχει τελειώσει κάνε τα εξής:

Φτιάξε έναν νέο κόμβο v που θα αντιστοιχεί στον χαρακτήρα αυτό;

Συνέδεσε κατάλληλα τον κόμβο αυτό στο δένδρο ως αριστερό ή δεξιό παιδί του τρέχοντος κόμβου (το αν το παιδί είναι αριστερό ή δεξιό εξαρτάται από το αν εκτελείται η πρώτη ή η δεύτερη ανακύκλωση, δηλαδή αν προχωράμε προς τα αριστερά ή αν έχει δημιουργηθεί το αριστερό υποδένδρο και ξεκινά η δημιουργία του δεξιού υποδένδρου);

Αν ο χαρακτήρας που διαβάστηκε είναι τελεστής, η MakeTree() καλείται αναδρομικά;

Υπόδειξη για την περίπτωση που χρησιμοποιηθεί στοίβα για τη δημιουργία του δένδρου: Για κάθε χαρακτήρα που διαβάζεται, δημιουργείται ένας νέος κόμβος στο δένδρο για το χαρακτήρα, ο οποίος τοποθετείται ως παιδί του τρέχοντος κόμβου. Αν ο χαρακτήρας είναι τελεστής, ένας δείκτης στον νέο κόμβο που δημιουργείται τοποθετείται σε στοίβα. Αν ο χαρακτήρας είναι μεταβλητή γίνεται pop() από τη στοίβα.

Άσκηση 3

α. Έστω 8 στοιχεία με κλειδιά τους αριθμούς 63, 30, 36, 31, 12, 50, 35, 14. Να τα εισάγετε διαδοχικά με την παραπάνω σειρά σε ένα αρχικώς άδειο ταξινομημένο δυαδικό δένδρο (δένδρο δυαδικής αναζήτησης). Στη συνέχεια να διαγράψετε τα στοιχεία με κλειδιά 30 και 63.

β. Θεωρήστε λεξικά που υλοποιούνται από διπλά-συνδεδεμένα, ταξινομημένα δυαδικά δένδρα. Έστω ότι θέλουμε επιπρόσθετα των Insert(), Delete() και LookUp() να υλοποιήσουμε τις ακόλουθες λειτουργίες στα λεξικά αυτά:

(i) findPredecessor(x): εύρεση του στοιχείου με το αμέσως μικρότερο κλειδί από αυτό του στοιχείου x. Η συνάρτηση επιστρέφει έναν δείκτη στον κόμβο με αυτό το στοιχείο ή τυπώνει την τιμή του στοιχείου αυτού.

(ii) findMaxGap(): Επιστροφή της διαφοράς $\max_{\forall\ x,\ y}$ (|x-y|). Η συνάρτηση παίρνει ως όρισμα έναν δείκτη στη ρίζα του δένδρου και επιστρέφει τη μέγιστη διαφορά μεταξύ οποιονδήποτε δύο κλειδιών των κόμβων του δένδρου.

Παρουσιάστε αλγορίθμους που να υλοποιούν τις παραπάνω λειτουργίες. Οι αλγόριθμοί σας θα πρέπει να έχουν την καλύτερη δυνατή χρονική πολυπλοκότητα.

Άσκηση 4

α. Παρουσιάστε μια συνάρτηση MaxElement() η οποία θα επιστρέφει το μέγιστο στοιχείο μιας λίστας παράλειψης (skip list).

β. Παρουσιάστε μια συνάρτηση DeleteMax(), η οποία θα διαγράφει το μεγαλύτερο στοιχείο μιας λίστας παράλειψης.

Υπόδειξη: Οι αλγόριθμοί σας θα πρέπει να επιτυγχάνουν καλή χρονική πολυπλοκότητα.