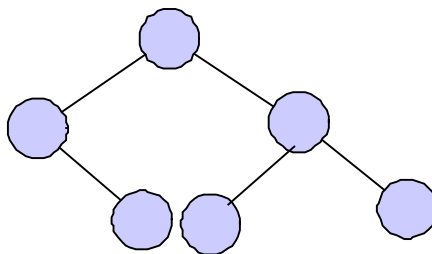


Πανεπιστήμιο Ιωαννίνων Σχολή Θετικών Επιστημών Τμήμα Μαθηματικών Τομέας Εφ. Μαθηματικών και Πληροφορικής	541 - ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ ΕΞΕΤΑΣΕΙΣ ΙΑΝΟΥΑΡΙΟΥ 2005
--	--

ΘΕΜΑ 1ον.

α. Δώσατε τους παρακάτω ορισμούς και από ένα παράδειγμα.
Δυαδικό δένδρο διερεύνησης, Σωρός, AVL δένδρο (Δένδρο ισοζυγισμένο κατά ύψος), Κώδικας Huffman.

β. Δώσατε τον ορισμό για το ελάχιστο μήκος εσωτερικών μονοπατιών και το ελάχιστο μήκος εξωτερικών μονοπατιών ενός δυαδικού δένδρου T με n εσωτερικούς κόμβους ($I(T)$ και $E(T)$ αντίστοιχα). Ποια είναι τα $I(T)$ και $E(T)$ για το παρακάτω δυαδικό δένδρο του 4(α); Αποδείξτε ότι $E(T)=I(T) + 2n$.



ΘΕΜΑ 2ον.

α. Από τις παρακάτω διασχίσεις επιλέξατε δύο και δώσατε το δυαδικό δένδρο αντιστοιχεί σε αυτές. Ποιες δύο διασχίσεις χρησιμοποίησατε και γιατί;

Ενθετική: EAGDHFBC
 Προθηματική: HGEADFCB
 Επιθηματική: AEDGBCFH

β. Περιγράψατε τα βήματα του αλγορίθμου για την μετατροπή μίας ενθετικής παράστασης σε επιθηματική με την χρήση στοίβας. Εφαρμόσατε τον αλγόριθμο στην παράσταση $(a+b*f)+c/(d-e*g)\#$

ΘΕΜΑ 3ον.

α. Ορίζουμε ένα νέο είδος δένδρου βαθμού m , ας το ονομάσουμε C - m δένδρο, με τις παρακάτω ιδιότητες.

1. Όλοι οι τερματικοί κόμβοι βρίσκονται σε δύο διαδοχικά επίπεδα $d-1$ και d .
2. Οι τερματικοί κόμβοι στο επίπεδο d δεν έχουν κενά μεταξύ τους και βρίσκονται στα αριστερά των τερματικών του επιπέδου $d-1$.
3. Όλοι οι εσωτερικοί κόμβοι έχουν βαθμό m εκτός ίσως από τον τελευταίο εσωτερικό (αυτός είναι ο δεξιότερος εσωτερικός στο προτελευταίο επίπεδο).

Να σχεδιάσετε μία δομή για την αποθήκευση C - m δένδρων χρησιμοποιώντας τον μικρότερο δυνατό χώρο. Εάν κάποιος κόμβος βρίσκεται στην θέση k της δομής, να δώσετε απλές σχέσεις για την θέση των παρακάτω.

Γονιός του k

Το i παιδί του k

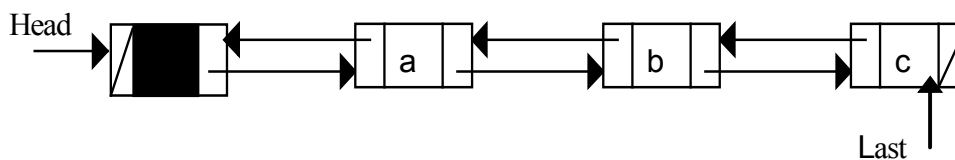
Ο αριστερός αδελφός του k

Ο δεξιός αδελφός του k

β. Έστω ότι θέλουμε να διασχίσουμε δένδρα βαθμού m με τον εξής τρόπο. επισκεπτόμαστε την ρίζα, μετά επισκεπτόμαστε τους κόμβους στο επίπεδο 1 από αριστερά προς τα δεξιά, μετά επισκεπτόμαστε τους κόμβους στο επίπεδο 2 από αριστερά προς τα δεξιά κ.ο.κ. Περιγράψατε πλήρως τα βήματα ενός αλγόριθμου για την διάσχιση αυτή. Ο αλγόριθμος σας πρέπει να είναι γραμμικός ως προς τον αριθμό των κόμβων του δένδρου. Πρέπει τα βήματα να είναι σαφή και να αναφέρατε τις δομές που θα χρησιμοποιήσετε.

ΘΕΜΑ 4ον.

α. Δώσατε σε JAVA τις κλάσεις κόμβος (NODE), διπλά συνδεδεμένη λίστα με κεφαλή (DLLIST) και μία κλάση ελέγχου (DLLISTTEST) στην οποία θα χρησιμοποιήσετε την διπλά συνδεδεμένη λίστα σε κάποια εφαρμογή.



Στις κλάσεις πρέπει να αναφέρετε τις μεταβλητές που χρησιμοποιείτε ενώ δεν χρειάζεται να αναφέρεται άλλες μεθόδους εκτός από τους κατασκευαστές στις δύο πρώτες και την main στην τελευταία όπου και θα πρέπει να δηλώνετε μία μεταβλητή DoubleLinkedList τύπου DLLIST.

β. Μελετήσατε το παρακάτω Java πρόγραμμα. Ποια είναι τα αντικείμενα της κλάσης TDM και πως έχουν υλοποιηθεί; Εξηγήσατε. Ποια είναι η έξοδος του main;

```

public class TDM
{
    // ΔΕΔΟΜΕΝΑ ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ
    int rows; // αριθμός γραμμών
    int [] element; // πίνακας για αποθήκευση στοιχείων

    // ΚΑΤΑΣΚΕΥΑΣΤΗΣ
    public TDM(int theRows)
    {if (theRows < 1)
        throw new IllegalArgumentException
            ("number of rows must be > 0");
        rows = theRows;
        element = new int [3 * rows - 2];
        for (int i = 0; i < 3 * rows - 2; i++)
            element[i] = 0;
    }

    // ΜΕΘΟΔΟΙ

    // ΕΛΕΓΧΟΣ ΔΕΙΚΤΩΝ
    void checkIndex(int i, int j)
    {if (i < 1 || j < 1 || i > rows || j > rows)
        throw new IndexOutOfBoundsException
            ("i = " + i + " j = " + j +
             " rows = " + rows + " cols = " + rows);
    }

    // ΑΝΑΚΛΗΣΗ ΤΙΜΗΣ - ΕΠΙΣΤΡΟΦΗ ΤΟΥ ΣΤΟΙΧΕΙΟΥ this(i,j)
    public int get(int i, int j)
    {checkIndex(i, j);
        switch (i - j)
        { case 1: return element[i - 2];
          case 0: return element[rows + i - 2];
          case -1: return element[2 * rows + i - 2];
          default: return 0;
        }
    }

    // ΑΠΟΘΗΚΕΥΣΗ ΝΕΑΣ ΤΙΜΗΣ - this(i,j) = newValue
    public void set(int i, int j, int newValue)
    {checkIndex(i, j);
        switch (i - j)
        { case 1: element[i - 2] = newValue; break;
          case 0: element[rows + i - 2] = newValue; break;
          case -1: element[2 * rows + i - 2] = newValue; break;
          default: if (!(newValue == 0))
                    throw new IllegalArgumentException
                        ("this matrix position must have a zero value");
        }
    }

    // ΕΚΤΥΠΩΣΗ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΜΕΛΟΥΣ element ΤΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΤΗΣ
    // ΚΛΑΣΗΣ
    // ΑΥΤΗ Η ΜΕΘΟΔΟΣ ΟΥΣΙΑΣΤΙΚΑ ΔΕΝ ΕΙΝΑΙ ΑΠΑΡΑΙΤΗΤΗ ΓΙΑ ΤΑ ΑΝΤΙΚΕΙΜΕΝΑ
    // ΤΗΣ ΚΛΑΣΗΣ ΚΑΙ ΧΡΗΣΙΜΟΠΟΙΕΙΤΑΙ ΜΟΝΟ ΓΙΑ ΤΗΝ ΤΡΕΧΟΥΣΑ ΕΡΩΤΗΣΗ
    public void print_elements()
    {for (int i = 0; i < 3 * rows - 2; i++)
        System.out.print(element[i]+" ");
        System.out.println();
    }
}

```

```

/** TEST ΠΡΟΓΡΑΜΜΑ */
public static void main(String [] args)
{
    TDM x = new TDM(10);

    // αποθήκευση μερικών τιμών
    x.set(3, 2, 150);
    x.set(5, 5, 100);
    x.set(9, 10, 50);
    x.set(8, 9, 200);

    // ανάκληση μερικών τιμών
    System.out.println(x.get(3, 2));
    System.out.println(x.get(5, 5));
    System.out.println(x.get(1, 1));
    System.out.println(x.get(9,10));
    System.out.println(x.get(8, 9));

    //ΕΚΤΥΠΩΣΗ ΤΩΝ ΣΤΟΙΧΕΙΩΝ ΤΟΥ ΜΕΛΟΥΣ element ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ
    x.print_elements();
}
}

```