

541 - ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ  
 ΕΞΕΤΑΣΕΙΣ ΣΕΠΤΕΜΒΡΙΟΥ 2005

ΘΕΜΑ 1ον.

α. Δώσατε ορισμούς και από ένα παράδειγμα για τα παρακάτω.

1. Ισοδύναμη δυαδικά δένδρα
2. Δένδρο βαθμού m
3. Δυαδικό δένδρο διερεύνησης
4. Ουρά

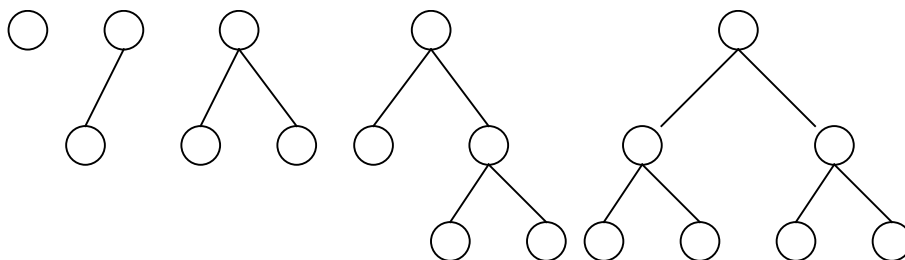
β. Σχεδιάσατε μία κατάλληλη δομή για την παράσταση και επεξεργασία ακεραίων πολυωνύμων  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ . Ο βαθμός των πολυωνύμων δεν είναι σταθερός ούτε γνωρίζουμε ένα μέγιστο όριο του βαθμού και τα πολυώνυμα μπορεί να έχουν μηδενικούς όρους τους οποίους δεν θέλουμε να αποθηκεύουμε.

ΘΕΜΑ 2ον.

α. Η αρίθμηση των δυαδικών δένδρων κατά Strahler ορίζεται ως εξής. Το κενό δένδρο έχει Strahler αριθμό 0. Εάν το δυαδικό δένδρο έχει αριστερό υπόδενδρο  $T_l$  και δεξιό  $T_r$ , τότε ο αριθμός Strahler  $S(T)$  του δένδρου  $T$  ορίζεται ως εξής.

$$S(T) = \begin{cases} \max[S(T_l), S(T_r)], & S(T_l) \neq S(T_r) \\ S(T_l) + 1, & S(T_l) = S(T_r) \end{cases}$$

Δώσατε τον αριθμό Strahler των παρακάτω δυαδικών δένδρων.



β. Δίδονται οι κάτωθι διασχίσεις ενός δυαδικού δένδρου και ζητείται να κατασκευαστεί το δυαδικό δέντρο.

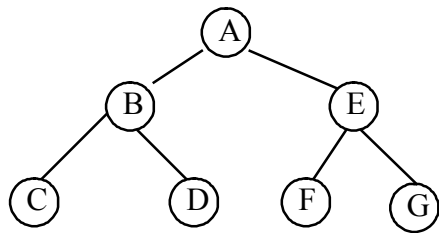
Προθηματική: EBDJCF AHIG

Ενθετική: JDCBEHAFIG

### ΘΕΜΑ 3ον.

α. Να εφαρμοστεί ο αλγόριθμος Huffman για τους χαρακτήρες της αλφαβήτου A,B,C,D,E,F,G με αντίστοιχες συχνότητες 0.1, 0.05, 0.15, 0.1, 0.05, 0.3, 0.25. Δώσατε το τελικό δένδρο που παράγει ο αλγόριθμος και τον τελικό κώδικα. Ποιο είναι το αναμενόμενο μήκος του παραγόμενου κώδικα; Είναι ο κώδικας που κατασκευάσατε άμεσα αποκωδικοποιήσιμος και γιατί;

β. Περιγράψατε με ψευδοκώδικα ένα μη αναδρομικό αλγόριθμο για την προθηματική διάσχιση ενός δυαδικού δένδρου με την χρήση στοίβας. Δείξατε την λειτουργία του αλγόριθμου στο παρακάτω δένδρο.



### ΘΕΜΑ 4ον.

α. Συμπληρώσατε τον παρακάτω σκελετό με κώδικα σε Java για την υλοποίηση της δομής της στοίβας με στοιχεία χαρακτήρες. Η στοίβα πρέπει να υλοποιηθεί με πίνακες.

```
// Στοίβα χαρακτήρων υλοποιημένη με πίνακα

class CharStack
{
    private int maxSize;           // μέγιστο μέγεθος
    private char[] stackArray;    // ο πίνακας
    private int top;              // η κορυφή

    // ο κατασκευαστής της στοίβας με max στοιχεία
    public CharStack(int max)
    {
        .....
    }

    // έλεγχος για κενή στοίβα
    public boolean isEmpty()
    {
        .....
    }

    // η μέθοδος push
    public void push(char j)
    {
        .....
    }
}
```

```

// η μέθοδος pop
public char pop()
{
    .....
}

// μέγιστο μέγεθος στοίβας
public int MaxCapacity()
{
    .....
}

// εκτύπωση
public void printStack()
{
    .....
}
}

```

β. Συμπληρώσατε τον παρακάτω σκελετό με κώδικα σε Java για την υλοποίηση μιας εφαρμογής η οποία δημιουργεί ένα αντίγραφο μιας στοίβας χρησιμοποιώντας μόνο την δομή της στοίβας που υλοποιήσατε παραπάνω και τις πράξεις της. Μπορείτε να κάνετε όποιες αλλαγές θεωρείτε αναγκαίες με την προϋπόθεση ότι το αντίγραφο μιας στοίβας δημιουργείται με χρήση μόνο των πράξεων της στοίβας που υλοποιήσατε παραπάνω.

```

// Η κλάση StackCopier χρησιμοποιεί την κλάση CharStack
// για την αντιγραφή μιας στοίβας χαρακτήρων

class StackCopier
{

    // Η μέθοδος StackCopy δημιουργεί ένα αντίγραφο της στοίβας S
    // και χρησιμοποιεί μόνο πράξεις της στοίβας
    public static CharStack StackCopy(CharStack S)
    {
        int stackSize = S.MaxCapacity(); // το μέγιστο μέγεθος της στοίβας

        .....

    }

    // κύρια μέθοδος
    public static void main(String[] args)
    {
        CharStack S1 = new CharStack(10); // κάποια στοίβα
        CharStack S2 = new CharStack(10); // το αντίγραφο
        // βάλτε μερικά στοιχεία στην S1
        .....
        // εκτύπωση της στοίβας S1
        System.out.println("Η αρχική στοίβα έχει τα στοιχεία:");
        .....
        // S2 είναι το αντίγραφο της S1
        S2 = .....;
        // εκτύπωση της στοίβας S2
        System.out.println("Το αντίγραφο της στοίβας έχει τα στοιχεία:");
        .....
    }
}

```