(Following Paper ID and Roll No. to be filled in your Answer Book)

PAPER ID : 214204   Roll No.

MCA.

(SEM. II) THEORY EXAMINATION 2013-14

DATA STRUCTURES AND FILE HANDLING

Time : 3 Hours   *Total Marks : 100*

Note :– Attempt questions from each sections as indicated.

SECTION–A

1. Attempt **all** parts :   **(2×10=20)**

    (a) Define Recursion.

    (b) Define Data Structure with suitable example.

    (c) Define Pop and Push operation in Stack.

    (d) What do you mean by the degree of node and degree of the tree.

    (e) Differentiate between Strictly and Almost Complete Binary Tree.

    (f) Draw binary tree of the following expression :

        (i) (A + B) * (c + d)

        (ii) (A + B + C) * (D + E + F)

    (g) Define B⁺ Tree.

    (h) Write the worst case complexity of Binary search.

    (i) Explain the concept of Devide & Conquer.

    (j) Define Spanning Tree.

SECTION–B

2. Attempt any **three** parts of the following : **(10×3=30)**

   (a) Write down both Iterative and Recursive version of Binary Search Algorithm. What is the time-complexity in both cases ?

   (b) Write a procedure SORT, which sorts a linked list without changing any values in information field of the nodes.

   (c) Write procedure of operations :

      (i) B-Tree Search

      (ii) B-Tree Insert

   (d) Illustrate the execution of HEAP SORT on the array.

      A = <6, 14, 3, 25, 2, 10, 20, 7 ,6>

   (e) Explain depth first search traversal algorithm of a graph.

SECTION–C

Note : Attempt any **five** of the following: **(10×5=50)**

3. Explain prims algorithm to find the minimum cost spanning tree of a weighted graph. Illustrate your algorithm with an example.

4. Write a program in C which sorts a list of n items using insertion sort method. Illustrate you algorithm with an example.

5. Write an algorithm to insert an item into a binary search tree.

6. Write functions in C that will :

   (i) Find height of a linked binary tree.

   (ii) Copy a linked binary tree.

7. Write C function to implement queues in a linear array with two indices 'front' and 'rear', such that when rear reaches the end of the array, all the items are moved to the front of the array.

8.   Explain B⁺ tree index files and B tree index files in detail.

9.   Write quick sort algorithm. Explain your algorithm taking suitable example. Analyze its running time.