

**K J SOMAIYA INSTITUTE OF MANAGEMENT STUDIES AND  
RESEARCH  
VIDYA NAGAR, VIDYA VIHAR, MUMBAI – 400 077**

**MCA-II : 2018-2021**

**Data Structures  
(Improvement Examination)**

**Max. Marks: 50**

**July, 2019**

**Duration: 3 hours**

**Instructions:**

Question No. I is **compulsory**

Answer **any two** Questions from the remaining three questions

Specify assumptions made wherever necessary

Figures on the right in brackets, indicate marks

**Q.1.a) Consider the structure below :**

[6 marks]

Struct Number

```
{  
    int num;  
    struct Number *next;  
};
```

Create a class called List which holds the head pointer of the structure and a count which holds the number of nodes in the list.

Create the insertAtBeg function to create the list till user desires and sortList function which sorts the list and Display function which displays the list.

Write a suitable client to demonstrate the three functions of the class.

**Q.1.b)** Explain the advantage of AVL trees over a BST with an example. Create an AVL tree for the following data coming in sequence. For each imbalance, mention the youngest ancestor, the case of imbalance and the subcase :

40, 39, 55, 26, 50, 72, 60, 90, 58

[8 marks]

**Q.1.c)** Give the step-by-step DFS and BFS traversal of the graph shown below :  
(No coding)

[6 marks]

**Q.2.a)** An array consists of the following set of numbers in the given sequence :

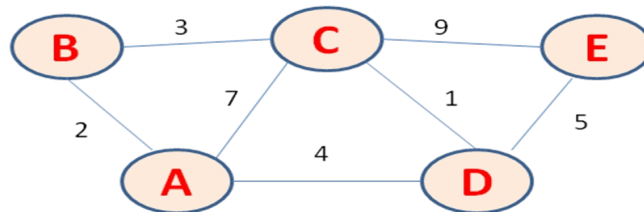
58, 63, 78, 10, 19, 81, 51, 25.37, 49

Show how the array would look like after two passes of : [6 marks]

1) SelectionSort

2) Insertion Sort

**Q.2.b)** Consider the graph below. Derive the minimum cost of the spanning tree using Kruskal's method. [8 marks]



**Q.3.a)** Consider the following set of numbers in an array : [7 marks]  
58, 63, 78, 10, 19, 81, 51, 25.37, 49

How will the array look like after 2 passes of :

i) Heapsort

ii) Quicksort

**Q.3.b)** Draw a Binary Tree given the following traversals. Also state the postorder traversal for the tree. [8 Marks]

Inorder : X Y T S M Z U R

Preorder : M S Y X T R Z U

**Q.4.a)** Define what's a complete tree and a nearly complete tree. Create a max heap using following keys coming in sequence : [7 marks]

42, 23, 74, 11, 65, 3, 94, 36, 99, 87

Also depict the tree after one deletion.

**Q.4.b)** Define the properties of a B-tree. Given the following set of numbers create a B-TREE of order 3 : [8 marks]

29, 8, 27, 99, 19, 32, 51, 41.

Now delete 41 from the tree and show the tree after the deletion.