S.E. (Computer Engineering) (First Semester)
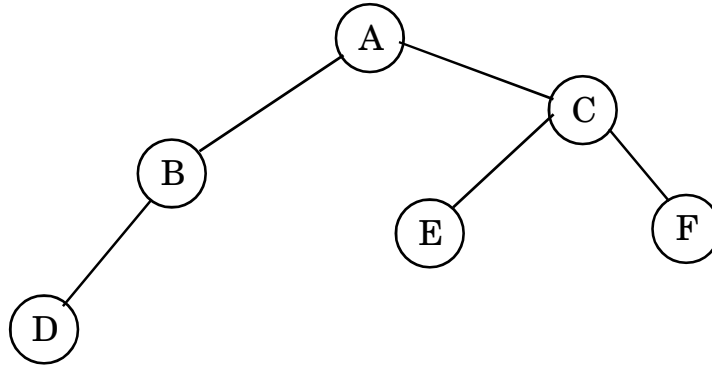## EXAMINATION, 2017
DATA STRUCTURES AND PROBLEM SOLVING
(2012 PATTERN)

Time : Two Hours   **Maximum Marks : 50**

N.B. :— (*i*)   Answer Q. **1** *or* Q. **2**, Q. **3** *or* Q. **4**, Q. **5** *or* Q. **6**, Q. **7** *or* Q. **8**.

   (*ii*)   Neat diagrams must be drawn whenever necessary.

   (*iii*)   Figrues to the right indicate full marks.

   (*iv*)   Use of calculator is allowed.

   (*v*)   Assume suitable data, if necessary.

1.   (*a*)   Represent the following binary tree using arrays :   [4]
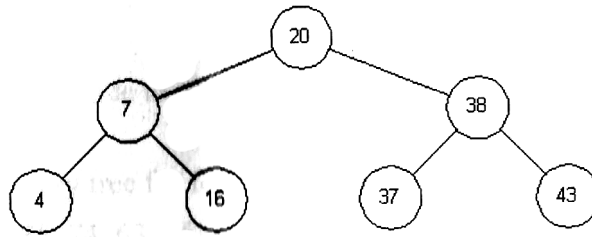

   (*b*)   Write the frequency count for the following code :   [4]

```
for(i=0;i<n;i++)
{
for(j=0;j<n;j++)
{
c[i][j]=a[i][j]+b[i][j];
}
}
```

(*c*) Sort the following data using quick sort in ascending order :

50, 30, 10, 90, 80, 20, 40, 70   [4]

Or

2. (*a*) What is Binary Search Tree ? Write inorder, preorder and postorder traversal of the following tree :   [4]

(*b*) Explain various Asymptotic notations.   [3]

(*c*) Explain the terms inorder successor and inorder predecessor with respect to threaded binary tree.   [2]

(*d*) Sort the following elements using radix sort in ascending order. Show all the steps.

35, 40, 7, 5, 50, 120, 99   [3]

3. (*a*) What is Hashing ? Explain hashing methods with example. [4]

(*b*) Explain topological sorting with example.   [4]

(*c*) Find Minimum Spanning Tree for the following graph using Kruskal's algorithm.   [4]
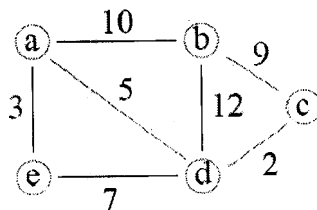
4. (*a*) Insert the following data in the hash table of size 10 using linear probing with chaining without replacement : [6]

131, 3, 4, 21, 61, 6, 71, 8, 9

(*b*) Explain various Graph storage structures. [6]

5. (*a*) Write algorithm to search an element in B tree. [6]

(*b*) Sort the following data in ascending order using heap sort : [4]

15, 10, 40, 25

(*c*) List and explain any *three* operations carried out on sequential file. [3]

6. (*a*) Write a pseudo C/C++ code to sort data using heap sort. [6]

(*b*) Construct 5-way binary tree for the following data : [7]

78, 21, 14, 11, 97, 85, 74, 63, 45, 42, 57, 20, 16, 19

7. (*a*) Write a parallel algorithm for odd-even merge sort. [7]

(*b*) Explain pointer doubling problem with example. [6]

8. (*a*) Explain prefix computation problem with example. [6]

(*b*) Write a parallel algorithm to perform addition of given numbers using complete binary tree method. [7]