

Roll No

EI/IC - 302**B.E. III Semester**

Examination, June 2016

Data Structure and Algorithms*Time : Three Hours**Maximum Marks : 70*

- Note:* i) Answer five questions. In each question part A, B, C is compulsory and D part has internal choice.
 ii) All parts of each question are to be attempted at one place.
 iii) All questions carry equal marks, out of which part A and B (Max. 50 words) carry 2 marks, part C (Max. 100 words) carry 3 marks, part D (Max. 400 words) carry 7 marks.
 iv) Except numericals, Derivation, Design and Drawing etc.

1. a) State the importance of dynamic programming.
 b) What are the two part of abstract data type? Explain.
 c) Consider the linear arrays AAA(5:50), BBB(-5:10) and CCC(18). Find the number of elements in each array.
 d) Explain the dynamic memory management with necessary methods.

OR

Explain in detail the steps involved in Top down Design.

2. a) Write the procedure for inserting an element in the list?
 b) What causes underflow of stack? How it could be avoided?
 c) Convert $((A+B)*C-(D-E))(F+G)$ to Postfix and Prefix notation.

[2]

- d) Define an efficient representation of two stacks in a given area of memory with n words and explain.

OR

Design an algorithm to reverse the linked list. Trace it with an example.

3. a) What is a almost complete binary tree?
 b) Explain the steps for symmetric order traversal.
 c) Define Hashing. What is meant by Perfect hash function?
 d) Construct an expression tree for the expression $A+(B-C)*D+(E*F)$.

OR

Create a Binary Search Tree for the following data and do in-order, Preorder and Post-order traversal of the tree.
50, 60, 25, 40, 30, 70, 35, 10, 55, 65, 5.

4. a) Define segment.
 b) What are the features of an efficient algorithm?
 c) What are the properties of an algorithm?
 d) Sort 20, 35, 40, 100, 3, 10, 15 using insertion sort.

OR

Explain merge sort with example.

5. a) Define adjacent nodes.
 b) What is meant by strongly connected in a graph?
 c) Define Dijkstra's algorithm.
 d) Explain the minimum spanning tree algorithms with an example.

OR

Explain Depth first and breadth first traversal.
