Seat No.

[5152]-563

S.E. (Comp.) (First Semester) EXAMINATION, 2017

DATA STRUCTURES AND ALGORITHNS

(2015 PATTERN)

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

1. (*a*) Show that $f(x) = 0(x^3)$ if function $f(x)$ is defined as $f(x) = 5x^3 + 6x^2 + 1$ [3]

    (*b*) Differentiate between linear and non-linear data structure with example. [3]

    (*c*) Explain divide and conquer strategy with example. Also comment on the time analysis. [6]

Or

2. (*a*) Explain fast Transpose of sparse matrix with suitable example. Discuss time complexity of fast transpose. [6]

    (*b*) Explain polynomial representation using arrays with suitable example. [3]

    (*c*) Derive recurrence relation to represent set of natural numbers giving remainder one when digvided by three. [3]

3. (*a*) Represent the following polynomial by using-generalized linked list : [3]

    (*a*, *b* (*c*, *d* (*e*, *g*,), *h*) (*f*))

    (*b*) Write an algorithm for postfix evaluation with suitable example. [6]

    (*c*) Write a pseudo C code to reverse singly linked list. [3]

P.T.O.

Or

4. (*a*) Convert the following prefix expression into postfix. $* + a - bc / - de + - fgh$ [3]

 (*b*) Write an algorithm to convert infix expression to postfix expression. [6]

 (*c*) Write an algorithm to delete intermediate node from Doubly linked list. [3]

5. (*a*) What is circular queue ? Explain the advantages of circular queue over linear queue. [6]

 (*b*) Write pseudo C/C++ code to represent queue as an ADT. [7]

Or

6. (*a*) Explain array implementation of priority queue with all basic operations. [6]

 (*b*) Write pseudo C/C++ code to implement circular queue using linked list. [7]

7. (*a*) Explain quick sort and sort the given list using quick sort : 39, 09, 81, 45, 90, 27, 72, 18 [6]

 (*b*) Write an algorithm for binary search. Derive recurrence relation and find out time complexity of the search. [7]

Or

8. (*a*) Explain heap sort and sort the given list using heap sort : 08, 03, 02, 11, 05, 14, 00, 02, 09, 04, 20. [6]

 (*b*) Write a short note an stability of sorting. Compare bubble sort, insertion sort and selection sort with *one* example and discuss time complexity. [7]