Total No. of Questions—**8**] [Total No. of Printed Pages—**3**

Seat No.	

[5057]-2052

S.E. (Computer Engineering) (First Semester)

EXAMINATION, 2016

DATA STRUCTURES AND ALGORITHMS

(2015 PATTERN)

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

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

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

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

1. (*a*) Define algorithm and its characteristics. [4]

(*b*) Write pseudo c/c++ code to perform simple transpose of sparse matrix. Discuss its time complexity. [6]

(*c*) Derive address calculation formula for one-dimensional array with one example. [2]

Or

2. (*a*) Explain asymptotic notations-Big-O, Theta and omega with one example of each. [6]

(*b*) Write pseudo c/c++ code to perform polynomial multiplication using arrays. [6]

P.T.O.

3. (*a*) Write pseudo c/c++ code to represent doubly linked list as an ADT. [6]

(*b*) Explain step-by-step conversion using stack for given infix expression to postfix expression : [6]

$$((a / (b - c + d)) * (e - a) * c$$

Or

4. (*a*) Write pseudo c/c++ code to implement stack as an ADT. [6]

(*b*) Write an algorithm to perform the following operations on singly linked list : [6]

(1) Reverse

(2) Sort.

5. (*a*) Write pseudo c/c++ code to represent deque and perform the following operations : [7]

(1) Create Deque

(2) Insert

(3) Delete

(4) Display.

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

Or

6. (*a*) Write pseudo c/c++ code to implement circular queue using arrays. [7]

(*b*) Explain applications of priority queue in detail. [6]

7. (*a*) Explain quick sort and sort the given list using quick sort : [6]

15, 08, 20, –4, 16, 02, 01, 12, 21, –2

(*b*) Write an algorithm for Fibonacci search and find out time complexity. [7]

Or

8. (*a*) Explain shell sort and sort the given list using shell sort : [6]

08, 03, 02, 11, 05, 14, 00, 02, 09, 04, 20

(*b*) Write short note on stability of sorting. Compare Heap sort and Quick sort with one example and discuss time complexity. [7]