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

Seat
No.

[5252]-563

S.E. (Computer Engineering) (I Semester) EXAMINATION, 2017
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 and explain the following terms :　　　　　　[3]

　　　　(*a*)　Data structure

　　　　(*b*)　ADT

　　　　(*c*)　Algorithm

　(*b*)　Give pseudo C/ C++ code to concatenate two strings.　[3]

　(*c*)　Explain the Greedy strategy with suitable example. Comment
　　　　on its time complexity.　　　　　　　　　　　　[6]

Or

2.　(*a*)　Define and explain the following terms :　　　　　　[4]

　　　　(*a*)　Linear data structure

　　　　(*b*)　Non-linear data structure

　　　　(*c*)　Time complexity

　　　　(*d*)　Space complexity

　(*b*)　What is sparse matrix ? Explain with suitable example. [2]

　(*c*)　Explain the Asymptotic notation Big O, Omega and Theta with
　　　　suitable example.　　　　　　　　　　　　　　　[6]

3. (*a*) Write a pseudo C/C++ code to delete intermediate node from singly linked list. [3]

(*b*) Explain Generalized linked list with example. [3]

(*c*) What is stack ? Write an ADT for stack. [6]

Or

4. (*a*) What is recursion ? Explain use of stack for recursion. [4]

(*b*) Explain the stepwise conversion using stack for the given infix expression to the postfix expression : [2]

A * (B + C) * D

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

5. (*a*) Define the following terms with example : [6]

(*a*) Dequeue

(*b*) Priority queue

(*c*) Linear queue

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

Or

6. (*a*) Explain linear queue and circular queue with suitable example. Give the advantages of circular queue over linear queue. [6]

(*b*) Explain priority queue. Give pseudo C/C++ code for array implementation of priority queue. [7]

7. (*a*) Sort the following numbers using Merge sort. [6]

55, 85, 45, 11, 34, 05, 89, 99, 67

Discuss its time complexity and space complexity.

(*b*) Explain sequential search and binary search with appropriate example and compare their time complexity and space complexity. [7]

Or

8. (*a*) Explain the algorithm of Quick sort with suitable example. Discuss its time complexity and space complexity. [6]

(*b*) Explain heap sort and sort the given list using heap sort : [7]
18, 13, 12, 22, 15, 24, 10, 16, 19, 14, 30.