Hall Ticket No | | | | | | | | | | | Question Paper Code: ACS002

INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)

Dundigal, Hyderabad - 500 043

MODEL QUESTION PAPER

Four Year B.TechII Semester End Examinations, May-2018

Regulations: IARE - R16

DATA STRUCTURES

(**Common to CSE / IT/ECE/EEE**)

Time: 3 hours **Max. Marks: 70**

Answer ONE Question from each Unit
All Questions Carry Equal Marks
All parts of the question must be answered in one place only

UNIT – I

1. a) For each of the following functions, find the best big-O function that bounds it. Justify youranswers. **[7M]**
   i. 5n+6-3n
   ii. n(n+1)/2
   iii. $n^2 + n \log n + 3n$
   iv. $\log n + \log(n^2)$
   v. log n!

   b) Given a list of integers 38, 6, 56, 23, 84, 9, 15, 27, 58, 42, 63, 11, 48. Compute the efficiency of Quick Sort algorithm for best, average and worst cases. **[7M]**

2. a) Compare the time complexities of various searching and sorting algorithms? **[7M]**

   b) Given a list of integers 9, 12, 23, 30, 35, 42, 55, 61, 71, 82, 99. Search for an element 35 in the sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty. **[7M]**

UNIT – II

3. a) Implement the basic stack operations PUSH, POP, DISPLAY using a list? **[7M]**

   b) Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. Find the conditions to detect queue full and queue empty by using the following conditions. **[7M]**
   i. Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT
   ii. Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR
   iii. Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT

iv. Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

4. a) Write aprogram to reverse a stack using recursion. Use the following ADT functions on Stack S: **[7M]**
isEmpty(S)
push(S)
pop(S)

b) Design a data structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack.Consider the following SpecialStack and return the minimum element in the current stack. **[7M]**
16 --> TOP
15
29
19
18

UNIT – III

5. a) Given a doubly linked list, write a function to sort the doubly linked list in increasing order using merge sort. **[7M]**

b) Given a singly linked list and a position, Write a program to delete a linked list node at the given position. **[7M]**

Input: position = 1, Linked List = 8->2->3->1->7
Output: Linked List = 8->3->1->7

Input: position = 0, Linked List = 8->2->3->1->7
Output: Linked List = 2->3->1->7

6. a) Write a function to implement the basic operations of a doubly linked list. **[7M]**

b) Given a singly linked list, write a program to check if the linked list is circular or not. A linked list is called circular if it not NULL terminated and all nodes are connected in the form of a cycle. **[7M]**

UNIT – IV

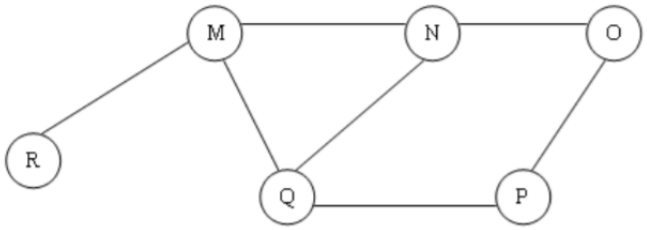
7. a) Define a binary search tree. Explain the operations of binary search tree with a suitable example? **[7M]**

b) Define a binary tree? Construct a binary tree given the pre-order traversal and in-order traversals as follows: **[7M]**
i) Pre-Order Traversal: G B Q A C K F P D E R H
ii) In-Order Traversal: Q B K C F A G P E D H R

[7M]

8. a) Write the procedure to be followed during infix to postfix conversion. Convert the following expression (2 + 3) – (4 / 5) ^ 7 from infix to postfix form.

b) The Breadth First Search algorithm has been implemented using the queue data structure. Discover breadth first search for the graph shown in Figure with starting node M **[7M]**



UNIT – V

9. a) Write how an AVL tree is different from Binary search tree and Create a AVL tree and binary searchtree for the given data: **[7M]**
56, 45,91,82,34,22,100,71,85,12

   b) Create a B-Tree of order 4 for the following data: **[7M]**
67,33,57,81,20,11,16,38,61,78

10. a) Define Hashing? Explain various collision resolution techniques? **[7M]**

   b) Define a binary tree? Construct a binary tree given the pre-order traversal and in-order traversals as follows: **[7M]**
      i. Pre-Order Traversal: G B Q A C K F P D E R H
      ii. In-Order Traversal: Q B K C F A G P E D H R

COURSE OBJECTIVES

The course should enable the students to:

S .No	Description
I	**Learn** the basic techniques of algorithm analysis.
II	**Demonstrate** searching and sorting algorithms and analyse their time complexities.
III	**Implement** linear data structures viz. stack, queue and linked list.
IV	**Demonstrate** non-linear data structures viz. tree and graph traversal algorithms.
V	**Study and choose** appropriate data structure to solve problems in real world.

COURSE LEARNING OUTCOMES

Students, who complete the course, will have demonstrated the ability to do the following:

S. No	Description	Blooms Taxonomy Level
ACS002.01	Understand algorithms and data structures in terms of time and space complexity of basic operations.	Understand
ACS002.02	Analyze a given problem; choose an appropriate data structure and an algorithm to solve the problem.	Understand
ACS002.03	Choose a suitable algorithm to organize the data in ascending or descending order.	Remember
ACS002.04	Understand the difference betweeniterative and recursion approaches to solve problems.	Remember
ACS002.05	Explore an algorithm to find the location of an element in a given list.	Understand
ACS002.06	Understand the usage of divide and conquer strategy in searching and sorting applications.	Understand
ACS002.07	Compare the time complexities of various searching and sorting algorithms.	Understand
ACS002.08	Understand the working principle of linear data structures and their real time applications.	Apply
ACS002.09	Organize the data in various linked representation format.	Apply
ACS002.10	Design and implement abstract data types for linear and non-linear data structures.	Understand
ACS002.11	Describe the concept of non-linear data structures viz. trees and graphs and their applications.	Understand
ACS002.12	Compare and Contrast the operations of binary search trees and AVL trees.	Understand
ACS002.13	Understand the concept of M-way search trees, operations and applications.	Understand
ACS002.14	List out different tree and graph traversal techniques.	Remember
ACS002.15	Understand the implementation of hashing using hash table and hash function.	Apply
ACS002.16	Describethe concept of collision and its resolving methodsin applications.	Apply
ACS002.17	Strengthen the knowledge of data structures and algorithms for employability.	Remember

MAPPING OF SEMESTER END EXAM TO COURSE LEARNINIG OUTCOMES

SEE Question No.		Course Learning Outcomes		Blooms Taxonomy Level
1	a	ACS002.01	Understand algorithms and data structures in terms of time and space complexity of basic operations.	Understand
	b	ACS002.03	Choose a suitable algorithm to organize the data in ascending or descending order.	Understand
2	a	ACS002.07	Compare the time complexities of various searching and sorting algorithms.	Understand
	b	ACS002.06	Understand the usage of divide and conquer strategy in searching and sorting applications.	Understand
3	a	ACS002.08	Understand the working principle of linear data structures and their real time applications.	Understand
	b	ACS002.08	Understand the working principle of linear data structures and their real time applications.	Apply
4	a	ACS002.08	Understand the working principle of linear data structures and their real time applications.	Understand
	b	ACS002.10	Design and implement abstract data types for linear and non-linear data structures.	Understand
5	a	ACS002.12	Understand the working principle of linear data structures and their real time applications.	Understand
	b	ACS002.09	Organize the data in various linked representation format.	Apply
6	a	ACS002.09	Organize the data in various linked representation format.	Understand
	b	ACS002.09	Organize the data in various linked representation format.	Understand
7	a	ACS002.12	Compare and Contrast the operations of binary search trees and AVL trees.	Understand
	b	ACS002.11	Describe the concept of non-linear data structures viz. trees and graphs and their applications.	Apply
8	a	ACS002.08	Understand the working principle of linear data structures and their real time applications.	Understand
	b	ACS002.14	List out different tree and graph traversal techniques.	Understand
9	a	ACS002.12	Compare and Contrast the operations of binary search trees and AVL trees.	Remember
	b	ACS002.13	Understand the concept of M-way search trees, operations and applications.	Remember
10	a	ACS002.15	Understand the implementation of hashing using hash table and hash function.	Apply
	b	ACS002.14	List out different tree and graph traversal techniques.	Apply

Signature of Course Coordinator

HOD, FRESHMAN ENGINEERING