



## Term Evaluation (Odd) Semester Examination September 2025

Roll no.....

Name of the Course

Semester: MTech. 1<sup>st</sup>

Name of the Paper: Data Structures and Algorithms

Paper Code: MCS-141

Time: 1:30 hour

Maximum Marks: 50

### Note:

- (i) All the questions are compulsory.
- (ii) Answer any of the two sub questions (a or b)
- (iii) Total marks for each question is 10 (ten).
- (iv) Each sub-question carries 10 marks.

Q1.

a) (10 Marks) (CO1)

- i. Solve the following recurrence relation using the recursion tree method.

$$T(n) = 2T(n/2) + n$$

and find the complexity.

- ii. Solve the following recurrence relation using Master Theorem.

$$T(n) = 3T(n/4) + n \log n$$

OR

- b) Define Big-O, Big-Ω, and Big-Θ notations. Explain with suitable examples how these notations help in analyzing algorithm performance.

Q2.

(10 Marks) (CO1)

- a. Write the recursive algorithm for solving the Towers of Hanoi problem. Also, derive the recurrence relation for the number of moves required to transfer n disks.

OR

- b. Explain the procedure for deletion from a Max Heap. Illustrate with an example how the heap property is maintained after deletion of the root element.

Q3.

(10 Marks) (CO1, 2)

- a. Write a recursive algorithm to implement quick sort. Also derive the recurrence relation and compute the complexity of the algorithm.

OR

- b. Convert the following infix expression to postfix expression

i.  $A + ((B * C) - (D / E ^ F)) + G$



## Term Evaluation (Odd) Semester Examination September 2025

ii.  $((A + B) * (C - D)) / (E + F * G)$

Q4.

(10 Marks) (CO2)

- a. What is a circularly linked list? Compare circular linked lists vs doubly linked lists in terms of traversal and memory usage.

OR

- b. Insert the following sequence into an empty Binary Search Tree (BST):

50, 30, 70, 20, 40, 60, 80

Draw the final BST and show the inorder, preorder, and postorder traversals of the tree.

Q5.

(10 Marks) (CO2)

- a) Differentiate between stack and queue in terms of:

- Data access order
- Implementation methods (array vs linked list)
- Applications

OR

- b) Write an algorithm to delete a node from a linked list. Consider all cases, i.e., deletion from beginning, deletion from between, deletion from end.