Seat No./Enrollment No.: _____

GUJARAT TECHNOLOGICAL UNIVERSITY
BACHELOR OF ENGINEERING – SEMESTER 5 – EXAMINATION – SUMMER 2026

Subject Code: 3150703 **Date: 06/06/2026**

Subject Name: Analysis and Design of Algorithms

Time: 02:30 PM TO 05:00 PM **Total Marks: 70**

Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**
4. **Simple and non-programmable scientific calculators are allowed.**

Q.1	**(a)**	Define domain, co-domain, and range of a function.	**03**
	(b)	Explain the amortized analysis technique with an example.	**04**
	(c)	Derive the time complexity of Heap Sort step by step with suitable justification.	**07**
Q.2	**(a)**	Perform Counting Sort on the input array A = [4, 2, 2, 8, 3, 3, 1]. Show all steps counting array, cumulative count, and the final sorted output.	**03**
	(b)	Solve the recurrence relation T(n) = 2T(n/2) + n using the Master Theorem.	**04**
	(c)	Derive the recurrence relation and solve it for Merge Sort using the Master Theorem.	**07**
		OR	
Q.2	**(c)**	Describe the Multiplying Large Integers Problem using Divide and Conquer.	**07**
Q.3	**(a)**	Explain the Principle of Optimality in Dynamic Programming with a suitable example.	**03**
	(b)	Explain the Huffman Coding algorithm with a suitable example and construct the Huffman Tree.	**04**
	(c)	Solve the 0/1 Knapsack problem using dynamic programming for the given data: Weights = [2, 3, 4, 5], Values = [3, 4, 5, 6], Capacity = 5.	**07**
		OR	
Q.3	**(a)**	State and explain the basic characteristics of problems that can be solved using Dynamic Programming.	**03**
	(b)	Describe the Knapsack Problem and mention why the Greedy approach works for the fractional version but not for the 0/1 version.	**04**
	(c)	For the two strings X = AGGTAB and Y = GXTXAYB, find the Longest Common Sub-sequence using Dynamic Programming.	**07**
Q.4	**(a)**	Explain the basic concepts of graphs with suitable examples. Differentiate between directed and undirected graphs.	**03**
	(b)	Perform Depth First Search (DFS) starting from vertex A for the graph: Vertices: A, B, C, D, E Edges: (A, B), (A, C), (B, D), (C, D), (D, E). Write the order of traversal.	**04**
	(c)	Find the Minimum Spanning Tree using Kruskal's algorithm for the following graph: Vertices = {A, B, C, D, E} Edges with weights: A-B: 2, A-C: 3, B-C: 1, B-D: 4, C-E: 5, D-E: 2.	**07**
		OR	

Q.4

(a) Explain the Breadth First Search (BFS) algorithm. Illustrate the traversal process using an example graph. **03**

(b) Find one possible topological ordering for the directed graph with edges: (5→2), (5→0), (4→0), (4→1), (2→3), (3→1). **04**

(c) Given items with weights (w) and profits (p): Item 1: (w=10, p=60), Item 2: (w=20, p=100), Item 3: (w=30, p=120), Capacity (W)=50. Solve using the Greedy method based on profit/weight ratio. **07**

Q.5

(a) Discuss the Eight Queens Problem in detail. Write the backtracking algorithm and demonstrate the solution process. **03**

(b) For the Rabin-Karp algorithm, given T = '31415926535', P = '26', base d = 10, and prime q = 13: Compute hash values of pattern and first few text substrings. Identify matching positions. **04**

(c) Describe the Travelling Salesman Problem (TSP). Why is it considered NP-hard? Discuss any one approximation algorithm for TSP. **07**

OR

Q.5

(a) Explain the Travelling Salesman Problem (TSP) using the Branch and Bound approach with a suitable example. **03**

(b) Given text T = 'AABAACAADAABAABA' and pattern P = 'AABA', trace the Knuth-Morris-Pratt algorithm execution step-by-step. Show all comparisons and matched positions. **04**

(c) Explain the class P SPACE and how it differs from P and NP. Provide suitable examples of P SPACE problems. **07**

*****