Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER–V (NEW) EXAMINATION – SUMMER 2019

Subject Code: 2150703 **Date: 06/06/2019**

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.**

			MARKS
Q.1	**(a)**	What is an algorithm? How it differ from flowchart?	**03**
	(b)	Give difference of dynamic programming and divide-and-conquer method.	**04**
	(c)	Explain Asymptotic notation. Arrange the growth rate of 2^n, n^2, 1, log n, n logn, 3^n and n in increasing order of growth.	**07**
Q.2	**(a)**	Differentiate greedy and dynamic programming.	**03**
	(b)	Find out the Θ-notation for the function: $f(n)=27n^2+16n$.	**04**
	(c)	What is recurrence? Explain recursion-tree method with suitable example.	**07**
		OR	
	(c)	Write the Master theorem. Solve following recurrence using it. (i) $T(n)=9T(n/3)+n$ (ii) $T(n)=3T(n/4)+n\lg n$	**07**
Q.3	**(a)**	Use Iteration method to solve recurrence $T(n) = T(n-1) + 1$, here $T(1)= \Theta(1)$.	**03**
	(b)	Explain general characteristics of greedy algorithms.	**04**
	(c)	Using dynamic programming find out the optimal sequence for the matrix chain multiplication of A_{4x10}, B_{10x3}, C_{3x12}, D_{12x20} and E_{20x7} matrices.	**07**
		OR	
Q.3	**(a)**	Write the best and worst running time of Insertion sort algorithm. Why it differ?	**03**
	(b)	What are the steps for dynamic programming? Explain principal of optimality.	**04**
	(c)	Determine LCS of {1,0,0,1,0,1,0,1} and {0,1,0,1,1,0,1,1,0}	**07**
Q.4	**(a)**	What is string-matching problem? Define valid shift and invalid shift.	**03**
	(b)	Define P, NP, NP-complete and NP-hard problems.	**04**
	(c)	Explain 0/1 knapsack using suitable example.	**07**
		OR	
Q.4	**(a)**	Write pseudo-code for Naïve-String-Matching algorithm.	**03**
	(b)	Define spanning tree and MST. How Krushkal's algorithm is different from Prim's algorithm.	**04**
	(c)	Explain fractional knapsack problem with example.	**07**
Q.5	**(a)**	Define graph, complete graph and connected graph.	**03**
	(b)	Differentiate BFS and DFS.	**04**
	(c)	Write and explain Dijkastra algorithm with example.	**07**
		OR	
Q.5	**(a)**	Explain "P = NP ?" problem.	**03**
	(b)	Write just steps for Backtracking and Branch-and-Bound algorithms.	**04**
	(c)	Explain travelling salesman problem. Prove that it is NP complete problem.	**07**

****