Seat No.: _____   Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

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

Subject Code:150703   **Date:31/05/2019**
Subject Name:Design And Analysis 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.

Q.1	**(a)**	Define Algorithm. Discuss factors affecting time complexity of an algorithm.	07
	(b)	Explain Big Oh (O), Omega (Ω) and Theta (θ) asymptotic notations.	07
Q.2	**(a)**	Apply merge sort algorithm on array A = {2,7,3,5,1,9,4,8}. What is time complexity of merge sort in worst case?	07
	(b)	Define Minimum Spanning Tree. Use Krushkal's algorithm to find Minimum Spanning Tree of given graph	07
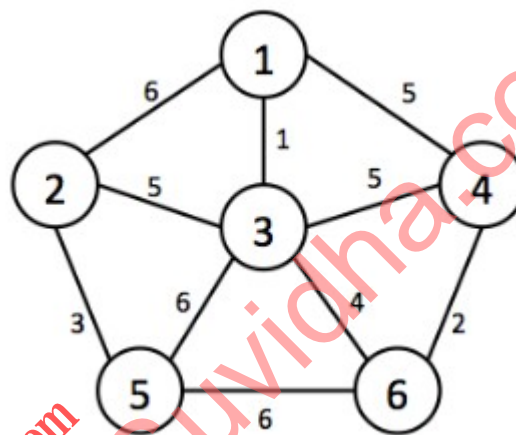
OR

	(b)	Discuss any two methods of amortized analysis in detail	07
Q.3	**(a)**	Write greedy algorithm for job scheduling problem. Derive its time complexity.	07
	(b)	Write divide and conquer algorithm to solve Exponential problem. Also solve 2^9 using same algorithm.	07

OR

Q.3	**(a)**	Obtain longest common subsequence using dynamic programming. Given A = "acabaca" and B = "bacac"	07
	(b)	Explain Depth First Search algorithm for a graph with example. Also explain Tree Edges, Back Edges and Cross Edges	07
Q.4	**(a)**	Solve making change problem using dynamic programming Given amount N=8, and denominations d = {1, 3, 5, 6}	07
	(b)	What is backtracking? How 4-Queen problem is solved using backtracking?	07

OR

Q.4	**(a)**	Sort given array A = {27, 46, 11, 95, 67, 32, 78} using insertion sort algorithm. Also perform best case and worst case analysis of insertion sort algorithm.	07
	(b)	How Rabin Karp algorithm performs string matching? Explain with example.	07
Q.5	**(a)**	Explain P Problem, NP Problem and NP Complete Problem.	07
	(b)	Write Naïve sting matching algorithm. Find its time complexity and perform sting matching for given pattern P = "ACD" Text T = "CACDACAACDAC"	07

1

OR

Q.5 **(a)** Explain in brief: Articulation Point, Directed Acyclic Graph, Recurrence Relations **07**

(b) Explain how to solve knapsack problem using greedy algorithms **07**

****