Seat No.: ________ Enrolment No.___________

GUJARAT TECHNOLOGICAL UNIVERSITY

BE - SEMESTER– V (New) EXAMINATION – WINTER 2019

Subject Code: 2150703 **Date: 25/11/2019**

Subject Name: Analysis and Design of Algorithms

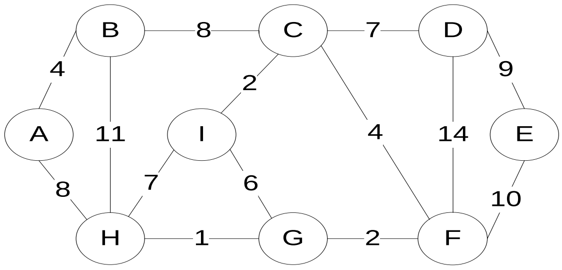
Time: 10:30 AM TO 01:00 PM **Total Marks: 70**

Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

			MARKS
Q.1	**(a)**	Find Omega (Ω) notation of function $f(n)=2n^2 + 6\ n * \lg n + 6n$.	**03**
	(b)	Define Big-oh and Theta notations with graph.	**04**
	(c)	Write sequential search algorithm and analyze it for worst case time complexity. Represent its time complexity using Big-oh (O) notation.	**07**
Q.2	**(a)**	Find upper bound of function $f(n)= \lg(n^2) + n^2 \lg n$.	**03**
	(b)	If $P(n) = a_0 + a_1 n + a_2 n^2 + \ldots\ldots + a_m n^m$ then prove that $P(n) = O(n^m)$. Here $a_0, a_1, a_2 \ldots.. a_m$ are constants and $a_m>0$.	**04**
	(c)	Solve following recurrence relation using suitable method and express your answer using Big-oh (O) notation. $T(n) = T(n/3) + T(2n/3) + \Theta(n)$	**07**
		OR	
	(c)	Solve following recurrence relation using suitable method and express your answer using Big-oh (O) notation. $T(n) = 2\ T(n/2) + n^2$	**07**
Q.3	**(a)**	If $T_1(n) = O(f(n))$ & $T_2(n) = O(g(n))$ then prove that $T_1(n) + T_2(n) = \max(O(g(n)), O(f(n)))$.	**03**
	(b)	Illustrate the working of the quick sort on input instance: 25, 29, 30, 35, 42, 47, 50, 52, 60. Comment on the nature of input i.e. best case, average case or worst case.	**04**
	(c)	Write greedy algorithm for activity selection problem. Give its time complexity. For following intervals, select the activities according to your algorithm. I_1 (1-3), I_2 (0-2), I_3 (3-6), I_4 (2-5), I_5 (5-8), I_6 (3-10), I_7 (7-9).	**07**
		OR	
Q.3	**(a)**	Arrange following growth rates in increasing order. $O(n^{1/4}), O(n^{1.5}), O(n^3 \lg n), O(n^{1.02}), \Omega(n^6), \Omega(n!), O(\sqrt{n}), O(n^{6/2}), \Omega(2^n)$	**03**
	(b)	Illustrate the working of the merge sort algorithm on input instance: 10, 27, 30, 88, 17, 98, 42, 54, 72, 95. Also write best case time complexity of merge sort algorithm.	**04**

(c) What is a minimum spanning tree? Draw the minimum spanning tree correspond to following graph using Prim's algorithm. **07**



Q.4 **(a)** What is Principle of Optimality? Explain it with example. **03**

(b) Consider the instance of the 0/1 (binary) knapsack problem as below with P depicting the value and W depicting the weight of each item whereas M denotes the total weight carrying capacity of the knapsack. Find optimal answer using greedy design technique. Also write the time complexity of greedy approach for solving knapsack problem. **04**

P = [40 10 50 30 60]   W = [80 10 40 20 90]   M = 110

(c) Find the optimal way of multiplying following matrices using dynamic programming. Also indicate optimal number of multiplications required. **07**

A:3 x 2, B: 2 x 5, C:5 x 4, D: 4 x 3, E: 3 x 3

OR

Q.4 **(a)** Explain depth first traversal using suitable example. **03**

(b) Explain Binomial Coefficient algorithm using dynamic programming. **04**

(c) Find the longest common subsequence for the following two sequences using dynamic programming. Show the complete process. **07**

X = 100101001

Y = 101001

Q.5 **(a)** Define P and NP problems. Also give example of each type of problem. **03**

(b) Draw the state space tree diagram for 4 Queen problem and also show the tree after applying backtracking. **04**

(c) Explain Rabin – Karp algorithm with example. What is expected running time of this algorithm? **07**

OR

Q.5 **(a)** Define NP-Complete and NP-Hard problems. Also give examples. **03**

(b) Explain the naive string matching algorithm. **04**

(c) State whether Hamiltonian problem is a NP-Complete problem? Justify your answer. **07**

****