# GUJARAT TECHNOLOGICAL UNIVERSITY
## BE- SEMESTER–V (NEW) EXAMINATION – WINTER 2020

**Subject Code:2150703**                                               **Date:22/01/2021**
**Subject Name:Analysis and Design of Algorithms**
**Time:10:30 AM TO 12:30 PM**                                  **Total Marks: 56**

**Instructions:**
1. Attempt any FOUR questions out of EIGHT questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.

| | | | |
|---|---|---|---|
| **Q.1** | **(a)** | Define O, $\Omega$, $\theta$ notations with example. | **03** |
| | **(b)** | Sort following functions in increasing order of running time for large values of n:  **n, $\log_2 n$, $2^n$, $n^2 \log n$, $n^3$** | **04** |
| | **(c)** | (i) What are the different parameters to analyze any algorithm? | **03** |
| | | (ii) Solve the following using Master's theorem: | |
| | | A. $T(n) = 2T(n/4) + 1$ | **04** |
| | | B. $T(n) = 3T(n/3) + n$ | |
| | | | |
| **Q.2** | **(a)** | Explain Master Theorem for all three cases. | **03** |
| | **(b)** | (i) What is the smallest value of n such that an algorithm whose running time is $100n^2$ is runs faster than an algorithm whose running time is $2^n$ on the same machine? | **04** |
| | | (ii) What is meaning of $T(n) = O(1)$. Explain with suitable example. | |
| | **(c)** | Given the four matrices $P_{5x4}$, $Q_{4x6}$, $R_{6x2}$, $T_{2x7}$. Find the optimal sequence for the computation of multiplication operation. | **07** |
| | | | |
| **Q.3** | **(a)** | Mention the parameters for finding suitable algorithm among many candidate algorithms. Justify parameter with suitable example. | **03** |
| | **(b)** | i.  Which version of algorithm is preferred when memory resources are limited, Iterative or recursive? Justify your answer. | **04** |
| | | ii.  An asymptotically fast algorithm running on Slow computer is better than asymptotically slow algorithm is running on fast computer for larger input size. (**True/False**) Justify your answer with supporting arguments. | |
| | **(c)** | Analyze Selection sort and Insertion Sort algorithms in best case and worst case scenarios. | **07** |
| | | | |
| **Q.4** | **(a)** | Merge sort algorithm have similar time complexity in best, average and worst case. (**True**/**False**). Justify your answer. | **03** |
| | **(b)** | Differentiate between greedy approach and Dynamic approach.. | **04** |
| | **(c)** | How the selection of pivot affects the performance of Quick Sort? Discuss all possible scenarios. | **07** |
| | | | |
| **Q.5** | **(a)** | How to solve knapsack problem using dynamic programming? | **03** |
| | **(b)** | Given two strings from 26 symbols set, X="BITTER", Y = "BUTTER" obtain the longest common subsequence. | **04** |

**(c)** Compare and contrast Branch and Bound and Backtracking Methods with **07** suitable example.

**Q.6** **(a)** Generate Huffman Code for symbols with probability as **03** $A_1(0.5), A_2(0.25), A_3(0.125), A_4(0.0625), A_5(0.0625)$.

**(b)** Find the all pair shortest path using Floyd-Warshall Algorithm for directed **04** graph shown below:
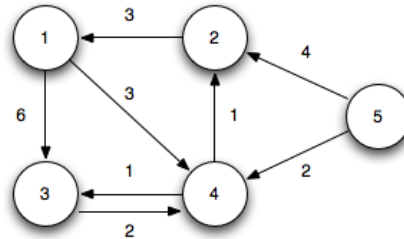


**(c)** How to solve 0-1 knapsack problem using dynamic programming? Consider **07** Items having Value(Rs.)={60,100,120} , Weight(KG)={10,20,30} respectively, Weight Capacity =50 KG.

**Q.7** **(a)** Define terms: Articulation Point, Isolated , Adjacency **03**
**(b)** Solve the following Task Assignment problem for minimization using **04** following cost matrix.(Cost matrix represents cost of Task T performed by Person P.

|    | T1 | T2 | T3 |
|----|----|----|----|
| P1 | 10 | 20 | 25 |
| P2 | 20 | 23 | 26 |
| P3 | 12 | 16 | 25 |

**(c)** Find minimum spanning tree for the following undirected weighted graph **07** using Kruskal's algorithm.
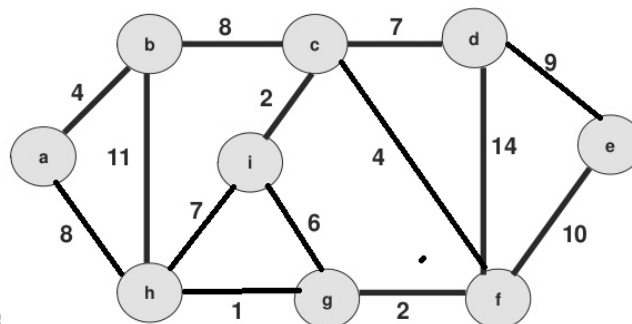


**Q.8** **(a)** What is the significance of Hashing in Rabin-Karp Pattern matching **03** algorithm?
**(b)** Draw the Finite automata which accepts String over 26 letter alphabet of **04** {A...Z} : **ACACAGA**
**(c)** Explain the concept of P, NP and NP-complete problem **07**

**\*\*\*\*\*\*\*\*\*\*\*\***