

## ALGORITHMS DESIGN AND ANALYSIS

**Paper Code: ETCS-301**

**Paper: Algorithms Design and Analysis**

<b>L</b>	<b>T/P</b>	<b>C</b>
<b>3</b>	<b>1</b>	<b>4</b>

**INSTRUCTIONS TO PAPER SETTERS:**

**MAXIMUM MARKS: 75**

1. Question No. 1 should be compulsory and cover the entire syllabus. This question should have objective or short answer type questions. It should be of 25 marks.
2. Apart from Question No. 1, rest of the paper shall consist of four units as per the syllabus. Every unit should have two questions. However, student may be asked to attempt only 1 question from each unit. Each question should be of 12.5 marks

***Objective:** The objective of this paper is to teach the students various problem solving strategies like divide and conquer, Greedy method, Dynamic programming and also the mathematical background for various algorithms. After doing this course, students will be able to select an appropriate problem solving strategies for real world problems. This will also help them to calculate the time, complexity and space complexity of various algorithms.*

### UNIT – I

Asymptotic notations for time and space complexity, Big-Oh notation,  $\Theta$  notation,  $\Omega$  notation, the little-oh notation, the little-omega notation, Recurrence relations: iteration method, recursion tree method, substitution method, master method (with proof), subtract and conquer master method(with proof), Data Structures for Disjoint Sets, Medians and Order statistics. Complexity analysis, Insertion sort, Merge Sort, Quick sort. Strassen's algorithm for Matrix Multiplications.

[T1][R1][R2][No. of Hrs. 10]

### UNIT – II

**Dynamic Programming:** Ingredients of Dynamic Programming, emphasis on optimal substructure , overlapping substructures, memorization. Matrix Chain Multiplication, Longest common subsequence and optimal binary search trees problems, 0-1 knapsack problem, Binomial coefficient computation through dynamic programming. Floyd Warshall algorithm.

[T1][T2][R1] [R3][No. of Hrs. 10]

### UNIT – III

**Greedy Algorithms:** Elements of Greedy strategy, overview of local and global optima, matroid, Activity selection problem, Fractional Knapsack problem, Huffman Codes, A task scheduling problem. Minimum **Spanning Trees:** Kruskal's and Prim's Algorithm, Single source shortest path: Dijkstra's and Bellman Ford Algorithm(with proof of correctness of algorithms).

[T1][T2][R4] [No. of Hrs. 10]

### UNIT – IV

**String matching:** The naïve String Matching algorithm, The Rabin-Karp Algorithm, String Matching with finite automata, The Knuth-Morris Pratt algorithm.

**NP-Complete Problem:** Polynomial-time verification, NP-Completeness and Reducibility, NP-Completeness Proof, NP –hard ,Case study of NP-Complete problems (vertex cover problem, clique problem).

[T1][R1] [No. of Hrs.: 10]

### Text Books:

[T1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, Clifford Stein, "Introduction to Algorithms", 3<sup>rd</sup> Ed., PHI, 2013.

[T2] Jon Klenberg, Eva Tardos, "Algorithm Design", Pearson Publications, 2014

### Reference Books:

[R1] Sara Basse, "introduction to Design & analysis", Pearson

[R2] Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Computer Algorithms/C++ "Second Edition, Universities Press.

[R3] A. V. Aho, J. E. Hopcroft, J. D. Ullman, "The Design and Analysis of Computer Algorithms", Pearson Publication, 2013.

[R4] Richard Neapolitan, "Foundations of Algorithms", Fifth Edition, Jones & Bartlett Learning