

# CSE408:DESIGN AND ANALYSIS OF ALGORITHMS

L:3 T:0 P:0 Credits:3

**Course Outcomes:** Through this course students should be able to

CO1 :: explain the basic techniques of analyzing the algorithms using space and time complexity, asymptotic notations

CO2 :: analyze various string matching algorithms and understand brute force algorithm design technique

CO3 :: understand divide and conquer algorithm design technique using various searching and sorting algorithms

CO4 :: define dynamic programming and greedy algorithm design technique and solve various all pair and single source shortest path problems

CO5 :: apply the backtracking method to solve some classic problems and understand branch and bound algorithm design technique

CO6 :: define various number theory problems and understand the basics concepts of complexity classes

## Unit I

**Foundations of Algorithm :** Algorithms, Fundamentals of Algorithmic Problem Solving:, Basic Algorithm Design Techniques, Analyzing Algorithm, Fundamental Data Structure:, Linear Data Structure, Graphs and Trees, Fundamentals of the Analysis of Algorithm Efficiency:, Measuring of Input Size, Units for Measuring Running Time, Order of Growth, Worst-Case, Best-Case, and Average-Case Efficiencies, Asymptotic Notations and Basic Efficiency Classes:,  $O(\text{Big-oh})$ -notation, Big-omega notation, Big-theta notation, Useful Property Involving the Asymptotic Notations, Using Limits for Comparing Orders of Growth

## Unit II

**String Matching Algorithms and Computational Geometry :** Sequential Search and Brute-Force String Matching, Closest-Pair and Convex-Hull Problem, Exhaustive Search, Voronoi Diagrams, Naive String-Matching Algorithm, Rabin-Karp Algorithm, Knuth-Morris-Pratt Algorithm

## Unit III

**Divide and Conquer and Order Statistics :** Merge Sort and Quick Sort, Binary Search, Multiplication of Large Integers, Strassen's Matrix Multiplication, Substitution Method for Solving Recurrences, Recursion-Tree Method for Solving Recurrences, Master Method for Solving Recurrence, Closest-Pair and Convex-Hull Problems by Divide and Conquer, Decrease and Conquer: Insertion Sort, Depth-First Search and Breadth-First Search, Connected Components, Topological Sort, Transform and Conquer: Presorting, Balanced Search Trees, Minimum and Maximum, Counting Sort, Radix Sort, Bucket Sort, Heaps and Heapsort, Hashing, Selection Sort and Bubble Sort

## Unit IV

**Dynamic Programming and Greedy Techniques :** Dynamic Programming: Computing a Binomial Coefficient, Warshall's and Floyd's Algorithm, Optimal Binary Search Trees, Knapsack Problem and Memory Functions, Matrix-Chain Multiplication, Longest Common Subsequence, Greedy Technique and Graph Algorithm: Minimum Spanning Trees, Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Code, Single-Source Shortest Paths, All-Pairs Shortest Paths, Iterative Improvement: The Maximum-Flow Problem, Limitations of Algorithm Power: Lower-Bound Theory

## Unit V

**Backtracking and Approximation Algorithms :** Backtracking: n-Queens Problem, Hamiltonian Circuit Problem, Subset-Sum Problem, Branch-and-Bound: Assignment Problem, Knapsack Problem, Traveling Salesman Problem, Vertex-Cover Problem and Set-Covering Problem, Bin Packing Problems

## Unit VI

**Number-Theoretic Algorithms and Complexity Classes :** Number Theory Problems: Modular Arithmetic, Chinese Remainder Theorem, Greatest Common Divisor, Optimization Problems, Basic Concepts of Complexity Classes- P, NP, NP-hard, NP-complete Problems

## Text Books:

1. INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHM by ANANY LEVITIN, PEARSON

## References:

1. INTRODUCTION TO ALGORITHMS by C.E. LEISERSON, R.L. RIVEST AND C. STEIN, THOMAS TELFORD LTD.

**References:**

2. THE DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS by A.V.AHO, J.E. HOPCROFT  
AND J.D.ULLMAN, PEARSON