

Design and Analysis of Algorithms

Course Title: Design and Analysis of Algorithms
Course No: CSC325
Nature of the Course: Theory + Lab
Semester: V

Full Marks: 60 + 20 + 20
Pass Marks: 24 + 8 + 8
Credit Hrs: 3

Course Description: This course introduces basic elements of the design and analysis of computer algorithms. Topics include asymptotic notations and analysis, divide and conquer strategy, greedy methods, dynamic programming, basic graph algorithms, NP-completeness, and approximation algorithms. For each topic, beside in-depth coverage, one or more representative problems and their algorithms shall be discussed.

Course Objectives:

- Analyze the asymptotic performance of algorithms.
- Demonstrate a familiarity with major algorithm design techniques
- Apply important algorithmic design paradigms and methods of analysis.
- Solve simple to moderately difficult algorithmic problems arising in applications.
- Able to demonstrate the hardness of simple NP-complete problems

Course Contents:

Unit 1: Foundation of Algorithm Analysis (4)

- 1.1. Algorithm and its properties, RAM model, Time and Space Complexity, detailed analysis of algorithms (Like factorial algorithm), Concept of Aggregate Analysis
- 1.2. Asymptotic Notations: Big-O, Big-Ω and Big-Θ Notations their Geometrical Interpretation and Examples.
- 1.3. Recurrences: Recursive Algorithms and Recurrence Relations, Solving Recurrences (Recursion Tree Method, Substitution Method, Application of Masters Theorem)

Unit 2: Iterative Algorithms (4)

- 12.1. Basic Algorithms: Algorithm for GCD, Fibonacci Number and analysis of their time and space complexity
- 12.2. Searching Algorithms: Sequential Search and its analysis
- 12.3. Sorting Algorithms: Bubble, Selection, and Insertion Sort and their Analysis

Unit 3: Divide and Conquer Algorithms (8)

- 13.1. Searching Algorithms: Binary Search, Min-Max Finding and their Analysis
- 13.2. Sorting Algorithms: Merge Sort and Analysis, Quick Sort and Analysis (Best Case, Worst Case and Average Case), Heap Sort (Heapify, Build Heap and Heap Sort Algorithms and their Analysis), Randomized Quick sort and its Analysis
- 13.3. Order Statistics: Selection in Expected Linear Time, Selection in Worst Case Linear Time and their Analysis.

Unit 4: Greedy Algorithms (6)

- 14.1. Optimization Problems and Optimal Solution, Introduction of Greedy Algorithms, Elements of Greedy Strategy.
- 14.2. Greedy Algorithms: Fractional Knapsack, Job sequencing with Deadlines, Kruskal's Algorithm, Prims Algorithm, Dijkstra's Algorithm and their Analysis
- 14.3. Huffman Coding: Purpose of Huffman Coding, Prefix Codes, Huffman Coding Algorithm and its Analysis

Unit 5: Dynamic Programming (8)

- 15.1. Greedy Algorithms vs Dynamic Programming, Recursion vs Dynamic Programming, Elements of DP Strategy
- 15.2. DP Algorithms: Matrix Chain Multiplication, String Editing, Zero-One Knapsack Problem, Floyd Warshwall Algorithm, Travelling Salesman Problem and their Analysis.
- 15.3. Memoization Strategy, Dynamic Programming vs Memoization

Unit 6: Backtracking (5)

- 16.1. Concept of Backtracking, Recursion vs Backtracking
- 16.2. Backtracking Algorithms: Subset-sum Problem, Zero-one Knapsack Problem, N-queen Problem and their Analysis.

Unit 7: Number Theoretic Algorithms (5)

- 7.1. Number Theoretic Notations, Euclid's and Extended Euclid's Algorithms and their Analysis.
- 7.2. Solving Modular Linear Equations, Chinese Remainder Theorem, Primality Testing: Miller-Rabin Randomized Primality Test and their Analysis

Unit 8: NP Completeness (5)

- 8.1. Tractable and Intractable Problems, Concept of Polynomial Time and Super Polynomial Time Complexity
- 8.2. Complexity Classes: P, NP, NP-Hard and NP-Complete
- 8.3. NP Complete Problems, NP Completeness and Reducibility, Cooks Theorem, Proofs of NP Completeness (CNF-SAT, Vertex Cover and Subset Sum)
- 8.4. Approximation Algorithms: Concept, Vertex Cover Problem, Subset Sum Problem

Laboratory Work:

This course can be learnt in effective way only if we give focus is given in practical aspects of algorithms and techniques discussed in class. Therefore student should be able to implement the algorithms and analyze their behavior. Students should:

- Implement comparison sorting algorithms and perform their empirical analysis.
- Implement divide-and-conquer sorting algorithms and perform their empirical analysis.
- Implement algorithms for order statistics and perform their empirical analysis.
- Implement algorithms by using Greedy, DP and backtracking paradigm
- Implement NP-complete problems and realize their hardness.

Recommended Books:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, "Introduction to algorithms", Third Edition.. The MIT Press, 2009.
2. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Computer Algorithms", Second Edition, Silicon Press, 2007.
3. Kleinberg, Jon, and Eva Tardos, "Algorithm Design", Addison-Wesley, First Edition, 2005