

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## Faculty of Engineering & Technology

### South Asian University, New Delhi, India

**Course Title: Design and Analysis of Algorithms**

**Course Code: CSE 201**

**Credits: 4.5**

**Pre-Requisites: Data Structures**

#### **COURSE OBJECTIVES:**

This course aims to equip students with a solid foundation in algorithm design and analysis. It focuses on developing a deep understanding of time and space complexity using asymptotic notation and recurrence relations. The students will learn and apply various algorithmic paradigms including divide and conquer, greedy, dynamic programming, and graph-based techniques to solve computational problems. Finally, students will be introduced to computational intractability through NP-completeness theory, enabling them to classify problems based on their computational difficulty.

#### **COURSE STRUCTURE:**

##### **Introduction to Algorithms**

Algorithms vs Programs, Characteristics of algorithms, Analysing algorithms: Time and space complexity, Asymptotic notation: Big-O, Big-Ω, Big-Θ, Small-o, Small-omega Recurrences and solution methods -Substitution, Recursion Tree, Master Theorem.

##### **Divide and Conquer Approaches**

General paradigm and recurrence relation, Binary Search, Merge Sort, Quick Sort, Matrix Multiplication (Strassen's Algorithm)

##### **Dynamic Programming**

Principle of Optimality and the distinction between bottom-up and top-down approaches in dynamic programming. Matrix Chain Multiplication, Longest Common Subsequence and Optimal Binary Search Tree,

##### **Greedy Algorithms**

Comparison of Greedy vs Dynamic with the help of Knapsack Problem and Activity Selection Problem, Huffman Coding for data compression using Greedy Approach.

##### **Graph Algorithms**

Graph representations, Breadth-First Search traversal and its applications - shortest path on an unweighted graph; Depth-First traversal and its applications - topological sorting of directed

acyclic graphs and identification of strongly connected components. Minimum Spanning Tree, Prim's and Kruskal's Algorithms for finding minimum spanning tree, Shortest Path problem using Dijkstra's Algorithm and Bellman-Ford Algorithm; Floyd-Warshall algorithm for all-pairs shortest path problem.

### **NP-Completeness and Algorithmic Strategies**

Theory of computational complexity – Definition of P, NP, NP-Hard & NP-Complete problems. Concept of Polynomial time verification, NP Completeness and Reducibility. NP Completeness Proofs. Algorithmic Strategies for NP-Complete Problems - Backtracking and Branch & Bound. Examples of NP complete problems.

### **SUGGESTED READINGS:**

#### **A. Text Books**

- a) Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.
- b) Kleinberg, Jon, and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.
- c) Ellis Horowitz, Sartaj Sahani, Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Orient Black Swan, 2nd Edition (2008).
- d) A .V. Aho, J . E . Hopcroft, J . D . Ulman "The Design & Analysis of Computer Algorithms", Addison Wesley, 1998.

#### **B. Web Resources**

- a) <https://nptel.ac.in/courses/106101060/>
- b) <https://ocw.mit.edu/courses/6-046j-introduction-to-algorithms-sma-5503-fall-2005/>

### **ASSESSMENT CRITERIA:**

- |                                  |                                  |
|----------------------------------|----------------------------------|
| 1. Quiz-1: 10%                   | 2. Quiz-2: 10%                   |
| 3. Mid-Semester Examination: 20% | 4. End-Semester Examination: 40% |
| 5. Practical Evaluation: 20%     |                                  |