

# Data structures and Algorithms

1. Unit 1 : Abstract Data Types
  - a. Linked Lists - Singly Linked List, Doubly Linked List, Circular Linked List
  - b. Stacks, Queues
  - c. Hash tables
  - d. Trees - Binary Trees, BST, AVL trees and height balancing, Red black trees.
  - e. Graphs
  - f. **Assignment 1**
2. Unit 2 : Sorting
  - a. Bubble sort, selection sort, insertion sort
  - b. Merge sort
  - c. Quick sort
  - d. Heap sort
  - e. Radix sort, Bucket sort
  - f. **Assignment 2**
3. Unit 3 : Algorithm Design, Paradigms and Analysis Methods
  - a. Divide and Conquer
  - b. Greedy Algorithms
  - c. Dynamic Programming
  - d. Amortized Analysis
  - e. **Assignment 3**
4. Unit 4 : Graph Algorithms
  - a. Graph Traversal - DFS, BFS, Level Order
  - b. Minimum Spanning Tree
  - c. Shortest distance - Single source shortest Distance, All pair shortest distance
  - d. Network Flow
  - e. **Assignment 4**
5. Unit 5 : Selected Algorithms and Computational Complexity
  - a. String Matching - Naive, Rabin-Karp algorithm , Knuth-Morris-Pratt algorithm
  - b. Number Theory - Greatest Common Divisor, Chinese Remainder
  - c. Matrix Multiplication - Strassen's algorithm
  - d. NP, NP-Complete, NP-Hard, Polynomial time verification
  - e. Approximation algorithms - Travelling Salesman Problem, Vertex Cover

## **Resources :**

Textbook : Introduction to Algorithms : Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein

Video Lectures : On specific topics in the syllabus from the following sources -

1. CS 170 : Efficient Algorithms and Intractable Problems, UC Berkeley
2. Introduction to Algorithms, MIT OCW

Assignment Questions : These will be selected from programming problems asked in software engineering interviews and ACM ICPC.

Source for practice problems :

1. Elements of Programming Interviews, Adnan Aziz, Amit Prakash, and Tsung-Hsien Lee
2. Cracking the Coding Interview, Gayle Laakmann McDowell
3. Topcoder, Leetcode

**Logistics :**

1. Each unit would comprise of reading the material on the specified topics, watching video lectures from the sources and solving practice problems from the specified sources.
2. At the end of each unit a programming assignment will be due. This programming assignment will contain problems from each of the topics covered in the unit. This will be due in 1 week after the reading for a particular unit is complete. The assignments will be tracked through a public Git repository.
3. The planned schedule for the programming assignments is as follows :
  - a. Assignment 1 : 9/28
  - b. Assignment 2 : 10/12
  - c. Assignment 3 : 10/26
  - d. Assignment 4 : 11/9