

# Data Structures using C

## 1. Introduction to data structures

- ✚ Concept
- ✚ Data type, Data object, ADT
- ✚ Need of Data Structure
- ✚ Types of Data Structure

## 2. Algorithm analysis

- ✚ 2.1 Algorithm – definition, characteristics
- ✚ 2.2 Space complexity, time complexity
- ✚ 2.3 Asymptotic notation (Big O, Omega  $\Omega$ )

## 3. Linear data structures

- ✚ 3.1 Introduction to Arrays - array representation
- ✚ 3.2 sorting algorithms with efficiency
- ✚ 3.3 bubble sort, Insertion sort, Merge sort, Quick Sort

## 4. Linked List

- ✚ 4.1 Introduction to List
- ✚ 4.2 Implementation of List – static & dynamic representation,
- ✚ 4.3 Types of Linked List
- ✚ 4.4 Operations on List
- ✚ 4.5 Applications of Linked List – polynomial manipulation
- ✚ 4.6 Generalized linked list – concept & representation

## 5. Stacks

- ✚ 5.1 Introduction
- ✚ 5.2 Representation-static & dynamic
- ✚ 5.3 Operations
- ✚ 5.4 Application - infix to postfix & prefix, postfix evaluation,
- ✚ 5.5 Recursion using implicit stack
- ✚ 5.6 Concept of Multiple stacks

## 6. Queues

- ✚ 6.1 Introduction
- ✚ 6.2 Representation -static & dynamic
- ✚ 6.3 Operations
- ✚ 6.4 Circular queue, DeQueue, priority queues
- ✚ 6.5 Concept of Multiple Queues

## 7. Trees

- ✚ 7.1 Concept & Terminologies
- ✚ 7.2 Binary tree, binary search tree
- ✚ 7.3 Representation – static & dynamic
- ✚ 7.4 Operations on BST – create. Insert, delete, traversals (preorder, inorder, postorder), counting
- ✚ leaf, non-leaf & total nodes
- ✚ 7.5 Application - Heap sort

✚ Height balance tree- AVL trees- Rotations

## **8. Graph**

- ✚ 8.1 Concept & terminologies
- ✚ 8.2 Graph Representation
- ✚ 8.3 Traversals – BFS & DFS
- ✚ 8.4 Applications – AOV network – topological sort
- ✚ AOE network – critical path
- ✚ Shortest path with implementation