

Data Structures & Algorithms using C and C++

1.Essential C and C++ Concepts

- Arrays Basics
- Structures
- Pointers
- Reference in C++
- Pointer to Structure
- Functions
- Parameter Passing Methods
- Structures and Functions
- Converting a C program to a C++ class

2.Required Setup for Programming

- Online C and C++ compiler
- Setup CodeBlocks and Settings
- Setup Dev-C++ and Settings
- Debugging using Dev-C++ and CodeBlocks
- Setup and Debugging using Visual Studio

3.Introduction

- Stack vs Heap Memory
- Physical vs Logical Data Structures
- Abstract Datatypes
- Time and Space Complexity

4.Recursion

- Generalizing Recursion
- Static and Global Variables in Recursion
- Head, Tail, and Tree Recursion
- Indirect Recursion
- Nested Recursion
- Fibonacci Series using Recursion

5.Array Representation

- Array Declaration
- Static vs Dynamic Arrays
- Increasing Array Size
- 2D Arrays
- Row and Column Major Formula for 2D Arrays
- Formulas for 3D & nD Arrays

6.Array ADT

- Array ADT
- Inserting in an Array
- Deleting from Array
- Linear Search
- Binary Search
- Get() Set() Avg() Max() functions on Array
- Merging and Sorting Arrays

7.Strings

- Finding Length of a String
- Validating a String
- Reversing a String

8.Matrices

- C++ class for Diagonal Matrix
- Lower Triangular Matrix Row and Column Major Mapping
- Lower Triangular Matrix in C and C++
- Triangular Matrices
- Symmetric Matrix

9.Sparse Matrix and Polynomial Representation

- Sparse Matrix Representation
- Create Sparse Matrix
- Sparse Matrix using C++
- Polynomial Representation
- Polynomial Evaluation
- Polynomial Addition
- Coding Polynomial

10.Linked List

- Dynamic Data Structure Linked List
- Display Linked List
- Recursive Display of Linked List
- Sum of All Elements in a Linked List
- Searching in a Linked List
- Deleting from Linked List
- Reversing a Linked List
- Concatenating 2 Linked Lists
- Merging 2 Linked Lists

11.Sparse Matrix and Polynomial Representation using Linked List

- Sparse Matrix using Linked List
- Polynomial Representation using Linked List

12.Stacks

- Stack using Array
- Implementation of Stack using Array
- Stack using Linked List
- Stack Operations using Linked List
- Parenthesis Matching
- Infix to Postfix Conversion
- Associativity and Unary Operators

13.Queues

- Queue ADT
- Queue using Single Pointer
- Queue using Two Pointers
- Implementing Queue using Array
- Drawback of Queue using Array
- Circular Queue
- Double Ended Queue DEQUEUE

14.Trees

- Terminology
- Number of Binary Trees using N Nodes
- Height vs Nodes in Binary Tree
- Internal Nodes vs External Nodes in Binary Tree
- Strict Binary Tree
- Height vs Node of Strict Binary Tree
- Internal vs External Nodes of Strict Binary Trees
- n-ary Trees
- Analysis of n-Ary Trees
- Full vs Complete Binary Tree

15.Binary Search Trees

- Searching in a Binary Search Trees
- Inserting in a Binary Search Tre
- Creating a Binary Search Tree
- Deleting from Binary Search Tree
- Program for Generating BST from Preorder

16.AVL Trees

- Inserting in AVL with Rotations
- General form of AVL Rotations
- Generating AVL Tree
- Deletion from AVL Tree with Rotations
- Height Analysis of AVL Trees

17.Search Trees

- 2-3 Trees
- 2-3-4 Trees
- Red-Black Trees
- Red-Black Trees vs 2-3-4 Trees
- Creating Red-Black Tree similar to Creating 2-3-4 Tree

18.Heap

- Inserting in a Heap
- Creating a Heap
- Deleting from Heap and Heap Sort
- Heap as Priority Queue

19.Sorting Technique

- Criteria used for Analysing Sorts
- Bubble Sort
- Insertion Sort
- Selection Sort
- Quick Sort
- Merging
- Recursive Merge Sort
- Count Sort
- Bin / Bucket Sort
- Radix Sort
- Shell Sort

20.Hashing Technique

- Chaining
- Linear Probing
- Quadratic Probing
- Double Hashing
- Hash Function Ideas

21.Graphs

- Directed and Undirected Graphs
- Breadth First Search
- Depth First Search
- Spanning Trees
- Prim's Minimum Cost Spanning Tree
- Kruskal's Minimum Cost Spanning Tree
- Disjoint Subsets

22.Asymptotic Notations

- Big Oh
- Omega
- Theta