GeeksforGeeks

DSA

Self Paced

DSA

Detailed
Course Syllabus

1) Introduction

- **Analysis of Algorithm**
  a) Background analysis through a Program and its functions.
- **Order of Growth**
  a) A mathematical explanation of the growth analysis through limits and functions.
  b) A direct way of calculating the order of growth
- **Asymptotic Notations**
  - Best, Average and Worst case explanation through a program.
- **Big O Notation**
  - Graphical and mathematical explanation.
  - Calculation
  - Applications at Linear Search
- **Omega Notation**
  - Graphical and mathematical explanation.
  - Calculation.
- **Theta Notation**
  - Graphical and mathematical explanation.
  - Calculation.
- **Analysis of common loops**
  - Single, multiple and nested loops
- **Analysis of Recursion**
  - Various calculations through Recursion Tree method
- **Space Complexity**
  - Basic Programs
  - Auxiliary Space
  - Space Analysis of Recursion
  - Space Analysis of Fibonacci number
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

2) Mathematics

- **Mathematics**
  - Count Digits
  - Palindrome Numbers
  - Factorial of Numbers
  - GCD of Two Numbers
  - LCM of Two Numbers
  - Check for Prime
  - Prime Factors
  - Sieve of Eratosthenes
  - Computing Power
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

3) Bit Magic

- **Bitwise Operators in C++**
  - Operation of AND, OR, XOR operators
  - Operation of Left Shift, Right Shift and Bitwise Not
- **Bitwise Operators in Java**
  - Operation of AND, OR
  - Operation of Bitwise Not, Left Shift
  - Operation of Right Shift and unsigned Right Shift
- **Problem(With Video Solutions): Check Kth bit is set or not**
  - Method 1: Using the left Shift.
  - Method 2: Using the right shift
- **Problem(With Video Solutions): Count Set Bits**
  - Method 1: Simple method
  - Method 2: Brian and Kerningham Algorithm
  - Method 3: Using Lookup Table
- **Problems(With Video Solutions):**
  - To check whether a number is a power of 2 or not
  - Odd occurrences in an array.
  - Two numbers having odd occurrences in an array.
  - Generate power set using bitwise operators.

- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

4) Recursion

- **Introduction to Recursion**
- **Applications of Recursion**
- **Writing base cases in Recursion**
  - Factorial
  - N-th Fibonacci number
- **Various problems on Recursion(With Video Solutions)**
  - Print n to 1
  - Print 1 to n
  - Tail Recursion
  - Checking Palindrome
  - Sum of digits
  - Rod cutting
  - Subsets of a set
  - Tower of Hanoi Problem
  - Josephus Problem
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

5) Arrays

- **Introduction and Advantages**
- **Types of Arrays**
  - Fixed-sized array
  - Dynamic-sized array
- **Operations on Arrays**
  - Searching
  - Insertions
  - Deletion

  - Arrays vs other DS
  - Reversing - Explanation with complexity
- **Problems(With Video Solutions)**
  - Left Rotation of the array by 1
  - Check if Sorted
  - Left Rotation of the array by D places
  - Leaders in an Array
  - Maximum Difference Problem
  - Frequencies in Sorted Array
  - Stock Buy and Sell Problem
  - Trapping Rainwater Problem
  - Maximum Consecutive 1s
  - Maximum Subarray Sum
  - Longest Even-Odd Subarray
  - Maximum Circular sum subarray.
  - Majority Element
  - Minimum Consecutive Flips
  - Sliding Window Technique
  - Prefix Sum Technique
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

6) Searching

- **Binary Search Iterative and Recursive**
- **Binary Search and various associated problems(With Video Solutions)**
  - Index of First Occurence in Sorted Array
  - Index of Last Occurence in Sorted Array
  - Count of occurrences of x in sorted element
  - Count of 1s in a binary sorted array
  - Find an element in sorted and rotated array
  - Peak element
  - Find an element in an infinite sized sorted array
  - The square root of an integer
- **Two Pointer Approach Problems(With Video Solutions)**
  - Find pair in an unsorted array which gives sum X
  - Find pair in a sorted array which gives sum X

  - Find triplet in an array which gives sum X
- **Problems(With Video Solutions)**
  - Median of two sorted arrays
  - Majority Element
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

7) Sorting

- **Implementation of C++ STL sort() function in Arrays and Vectors**
  - Time Complexities
- **Sorting in Java**
- **Arrays.sort() in Java**
- **Collection.sort() in Java**
- **Stability in Sorting Algorithms**
  - Examples of Stable and Unstable Algos
- **Bubble Sort**
- **Selection Sort**
- **Insertion Sort**
- **Merge Sort**
- **Problems(With Video Solutions)**
  - Intersection of 2 sorted arrays
  - Union of 2 sorted arrays
  - Count Inversions in arrays
- **Partitions(With Video Solutions)**
  - Naive
  - Lomuto
  - Hoare
- **Quick Sort**
  - Using Lomuto and Hoare
  - Time and Space analysis
  - Choice of Pivot and Worst case
  - Tail call elimination
- **Problems(With Video Solutions)**
  - Kth Smallest element
  - Chocolate Distribution Problem
  - Sorting arrays with 2 and3 types of elements

    - Merge Overlapping Intervals
    - Meeting the Maximum Guests
- **Heap Sort**
- **Cycle Sort**
- **Counting Sort**
- **Radix Sort**
- **Bucket Sort**
- **Overview of Sorting Algorithms**
- **Practice Problems**
    - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

8) Matrix

- **Introduction to Matrix in C++ and Java**
- **Multidimensional Matrix**
- **Pass Matrix as Argument**
- **Printing matrix in a snake pattern**
- **Transposing a matrix**
- **Rotating a Matrix**
- **Check if the element is present in a row and column-wise sorted matrix.**
- **Boundary Traversal**
- **Spiral Traversal**
- **Matrix Multiplication**
- **Search in row-wise and column-wise Sorted Matrix**
- **Practice Problems**
    - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

9) Hashing

- **Introduction and Time complexity analysis**

- **Application of Hashing**
- **Discussion on Direct Address Table**
- **Working and examples on various Hash Functions**
- **Introduction and Various techniques on Collision Handling**
- **Chaining and its implementation**
- **Open Addressing and its Implementation**
- **Chaining V/S Open Addressing**
- **Double Hashing**
- **C++**
    - Unordered Set
    - Unordered Map
- **Java**
    - HashSet
    - HashMap
- **Problems(With Video Solutions):**
    - Count Distinct Elements
    - Count of the frequency of array elements
    - The intersection of two arrays
    - Union of two unsorted arrays
    - Pair with given sum in an unsorted array
    - Subarray with zero-sum
    - Subarray with given sum
    - Longest subarray with a given sum
    - Longest subarray with an equal number of 0's and 1's
    - Longest common span with the same sum in a binary array
    - Longest Consecutive Subsequence
    - Count Distinct elements in every window
    - More than n/k Occurences
    - Optimized More than n/k Solution
- **Practice Problems**
    - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

10) Strings

- **Discussion of String DS**
- **Strings in CPP**
- **Strings in Java**

- **Problems(With Video Solutions):**
  - Given a string, check if they are an anagram of each other.
  - Given a string, find the leftmost character that repeats.
  - Given a string, find the leftmost character that does not repeat.
  - Given a string, find the lexicographic rank of it in O(n) time.
  - Implementation of the previously discussed lexicographic rank problem.
  - Given a text string and a pattern string, find if a permutation of the pattern exists in the text.
  - Given two strings, check if they are rotations of each other or not.
  - Various Pattern Searching Algorithms.
  - Palindrome Check
- **Rabin Karp Algorithm**
- **KMP Algorithm**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

11) Linked List

- **Introduction**
  - Implementation in CPP
  - Implementation in Java
  - Comparison with Array DS
- **Doubly Linked List**
- **Circular Linked List**
- **Loop Problems**
  - Detecting Loops
  - Detecting loops using Floyd cycle detection
  - Detecting and Removing Loops in Linked List
- **Problems(With Video Solutions):**
  - Middle of Linked List
  - Nth node from the end of linked list
  - Deleting a Node without accessing Head pointer of Linked List
  - An iterative method to Reverse a linked list
  - Recursive method to reverse a linked list
  - Reverse in group of size k
  - Recursive Traversal in a Singly Linked List

  - Segregating even-odd nodes of linked list
  - The intersection of two linked list
  - Pairwise swap nodes of linked list
  - Clone a linked list using a random pointer
  - LRU Cache Design
  - Merge two Sorted Linked Lists
  - Palindrome Linked List
  - Recursive Traversal in a Singly Linked List
  - Remove Duplicates from a Sorted Singly Linked List
  - Sorted Insert in a Singly Linked List
  - Reverse a Doubly Linked List
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

12) Stack

- **Understanding the Stack data structure**
- **Applications of Stack**
- **Implementation of Stack in Array and Linked List**
  - In C++
  - In Java
- **Problems(With Video Solutions):**
  - Balanced Parenthesis
  - Two stacks in an array
  - K Stacks in an array
  - Stock span problem with variations
  - Previous Greater Element
  - Next Greater Element
  - Largest Rectangular Area in a Histogram
- **Understanding getMin() in Stack with O(1)**
- **Infix, Prefix and Postfix Introduction**
  - Infix to Postfix (Simple Solution)
  - Infix to Postfix (Efficient Solution)
  - Evaluation of Postfix
  - Infix to Prefix (Simple Solution)

    - Infix to Prefix (Efficient Solution)
    - Evaluation of Prefix
- **Practice Problems**
    - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

13) Queue

- **Introduction and Application**
- **Implementation of the queue using array and LinkedList**
    - In C++ STL
    - In Java
    - Stack using queue
- **Problems(With Video Solutions)**
    - Reversing a Queue
    - Generate numbers with given digits
    - First Circular Tour
- **Practice Problems**
    - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

14) Deque

- **Introduction and Application**
- **Implementation**
    - In C++ STL
    - In Java
- **Problems(With Video Solutions)**
    - Maximums of all subarrays of size k
    - ArrayDeque in Java
    - Design a DS with min max operations
- **Practice Problems**

- This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

15) Tree

- **Introduction**
  - Tree
  - Application
  - Binary Tree
  - Tree Traversal
- **Implementation of:**
  - Inorder Traversal
  - Preorder Traversal
  - Postorder Traversal
  - Level Order Traversal (Line by Line)
  - Tree Traversal in Spiral Form
- **Problems(With Video Solutions):**
  - Size of Binary Tree
  - Maximum in Binary Tree
  - Height of Binary Tree
  - Print Nodes at K distance
  - Print Left View of Binary Tree
  - Children Sum Property
  - Check for Balanced Binary Tree
  - Maximum Width of Binary Tree
  - Convert Binary Tree to Doubly Linked List
  - Construct Binary Tree from Inorder and Preorder
  - Tree Traversal Spiral Form
  - The diameter of a Binary Tree
  - LCA problem with an efficient solution
  - Burn A Binary Tree from a Leaf
  - Count Nodes in a complete Binary Tree
  - Serialize and Deserialize a Binary tree
  - Iterative Inorder Traversal
  - Iterative Preorder Traversal (Simple)
  - Iterative Preorder Traversal (Space Optimized)
- **Practice Problems**

  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

16) Binary Search Tree

- ✓ **Background, Introduction and Application**
- ✓ **Implementation of Search in BST**
  - In CPP
  - In Java
- ✓ **Insertion in BST**
  - In CPP
  - In Java
- **Deletion in BST**
  - In CPP
  - In Java
- **Floor in BST**
  - In CPP
  - In Java
- **Self Balancing BST**
- ✓ **AVL Tree**
- **Red Black Tree**
- **Set in C++ STL**
- **Map in C++ STL**
- **BST Introduction**
- **TreeSet in java**
- **TreeMap in Java**
- **Problems(With Video Solutions):**
  - The ceiling of a key in BST
  - Ceiling on the left side in an array
  - Find Kth Smallest in BST
  - Check for BST
  - Fix BST with Two Nodes Swapped
  - Pair Sum with given BST
  - Vertical Sum in a Binary Tree
  - Vertical Traversal of Binary Tree
  - Top View of Binary Tree
  - Bottom View of Binary Tree
- **Practice Problems**

- This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

17) Heap

- **Introduction & Implementation**
- **Binary Heap**
  - Insertion
  - Heapify and Extract
  - Decrease Key, Delete and Build Heap
- **Heap Sort**
- **Priority Queue in C++**
- **PriorityQueue in Java**
- **Problems(With Video Solutions):**
  - Sort K-Sorted Array
  - Buy Maximum Items with Given Sum
  - K Largest Elements
  - Merge K Sorted Arrays
  - Median of a Stream
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

18) Graph

- **Introduction to Graph**
- **Graph Representation**
  - Adjacency Matrix
  - Adjacency List in CPP and Java
  - Adjacency Matrix VS List
- **Breadth-First Search**
  - Applications
- **Depth First Search**
  - Applications

- **Problems(With Video Solutions):**
  - Shortest Path in an Unweighted Graph
  - Detecting Cycle
    - In the Undirected Graph
    - In the Directed Graph
  - Topological Sorting
    - Kahn's BFS Based Algorithm
    - DFS Based Algorithm
- **Shortest Path in Directed Acyclic Graph**
- **Prim's Algorithm/Minimum Spanning Tree**
  - Implementation in CPP
  - Implementation in Java
- **Dijkstra's Shortest Path Algorithm**
  - Implementation in CPP
  - Implementation in Java
- **Bellman-Ford Shortest Path Algorithm**
- **Kruskal's Algoritm**
- **Kosaraju's Algorithm**
- **Articulation Point**
- **Bridges in Graph**
- **Tarjan's Algorithm**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

19) Greedy

- **Introduction**
- **Activity Selection Problem**
- **Fractional Knapsack**
- **Job Sequencing Problem**
- **Huffman Coding**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

20) Backtracking

- **Concepts of Backtracking**
- **Rat In a Maze**
- **N Queen Problem**
- **Sudoku Problem**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

21) Dynamic Programming

- **Introduction**
- **Dynamic Programming**
  - Memoization
  - Tabulation
- **Problems(With Video Solutions):**
  - Longest Common Subsequence
  - Coin Change Count Combinations
  - Edit Distance Problem
    - Naive Approach
    - DP Approach
  - Longest Increasing Subsequence Problem
    - Naive Approach
    - Efficient Approach
  - Maximum Cuts
  - Minimum coins to make a value
  - Minimum Jumps to reach at the end
  - 0-1 knapsack problem
    - Naive Approach
    - Efficient Approach
  - Optimal Strategy for a Game
  - Variation of Longest Common Subsequence
  - Variation of Longest Increasing Subsequence
  - Egg Dropping Problem
  - Count BST with nkeys
  - Maximum Sum with No Consecutive
  - Subset Sum Problem

  - Matrix Chain Multiplication
  - Palindrome Parititioning
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

22) Trie

- **Introduction**
  - Representation
  - Search
  - Insert
  - Delete
- **Count Distinct Rows in a Binary Matrix**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

23) Segment Tree

- **Introduction**
- **Construction**
- **Range Query**
- **Update Query**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

24) Disjoint Set

- **Introduction**
- **Find and Union Operations**
- **Union by Rank**
- **Path Compression**

- **Kruskal's Algorithm**
- **Practice Problems**
  - This track contains many practice problems for the users which are considered important and must-do as far as Data Structure and Algorithm is concerned.

25) Projects

- **Sudoku Solver**

  Write a program to solve a Sudoku puzzle by filling the empty cells.
  A sudoku solution must satisfy all of the following rules

- **Shortest Path Finder**

  The problem of finding the shortest path between two intersections on a road map may be modeled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment.

- **Tic Tac Toe**

  A game in which two players alternately put Xs and Os in compartments of a figure formed by two vertical lines crossing two horizontal lines and each tries to get a row of three Xs or three Os before the opponent does

- **N Queen Visualizer**

  Visualization of solving the N-Queens puzzle using recursive algorithm. The N-Queens puzzle is the problem of placing N chess queens on an N×N chessboard so that no two queens threaten each other