DSA LIVE
FOR
WORKING PROFESSIONAL

Detailed Course Syllabus
Week 01

Class 1:
- **Analysis of Algorithm**
- **Mathematics**
- **Bit Manipulation**
  - Asymptotic analysis
  - Time and Space Complexity
  - Master Theorem
  - Bitwise Operators (Bitwise AND, Bitwise OR, Bitwise XOR, Left Shift, Right Shift, etc)

**Practice Problems**
GCD and LCM
Iterative Power
Computing Power
Prime factorization
Prime Numbers (Sieve Algorithms)
Generate Power Set

Class 2:
- **ARRAYS**
  - Arrays - Introduction and Advantages
  - Types of Arrays
  - Operations of Arrays - Searching, Insertion, Deletion,
    - Sliding Window Technique

**Practice Problems**
Largest Element in an Array
Leaders in an Array problem
Reverse an Array
Maximum subarray sum

Week 02

Class 1:
- **RECURSION**
- **BACKTRACKING**
  - Introduction to Recursion
  - Writing Base Cases in Recursion
  - Tail Recursion
  - Introduction to Backtracking
Practice Problems
Print 1 to N Using Recursion
Sum of Digits Using Recursion
Rope Cutting Problem
Generate Subsets
Tower of Hanoi
Josephus Problem
Rat in a Maze Problem

Class 2: - SEARCHING
- Linear Search
- Binary Search - Iterative and Recursive Approach
- Analysis of Binary Search
- Two Pointer Approach

Practice Problems
Index of first Occurrence in Sorted
Count 1s in a Sorted Binary Array
Square root of a number
Search in an Infinite sized array
Search in a sorted rotated array
Triplet in a Sorted Array
Allocate Minimum Pages (Binary Search)

Week 03

Class 1: - SORTING
- Overview of sorting algorithm
- Sorting Algorithms like Insertion, Bubble, Selection, Merge and Quick Sort
- Stability of Sorting Algorithms

Practice Problems
Minimum Difference in an Array
Chocolate Distribution Problem
Union of two Sorted Arrays
Kth Smallest Element
Sort an Array with two/three types of element
CONTENTS

Class 2:  
- MATRIX
- HASHING
  - Multidimensional Array
  - Passing 2D arrays as argument
  - Hashing Introduction and Application, Time Complexity Analysis
  - Collision Handling
  - Hashing Function
  - Unordered Set and Unordered Map
  - HashSet and HashMap

Practice Problems
Transpose of a Matrix
Matrix in Snake Pattern
Spiral Traversal of a Matrix
Count Distinct Elements
Frequencies of Array Elements
Intersection and Union of two Array Elements
Subarray with given Sum
Count Distinct Elements in Every Window
More than n/k Occurrence (with O(nk) solution)

Week 04

Class 1:  
- STRINGS
  - Introduction to Strings
  - Overview of Pattern Searching Algorithm
  - Naive and Improved Naive Pattern Searching
  - Rabin Karp Algorithm
  - KMP Algorithm (Constructing LPS Array and Complete Algorithms)

Practice Problems
Palindromic Check
Reverse words in a string
Check for Anagram
Check if Strings are Rotations
Anagram Search
Lexicographic Rank of a String
Longest Substring with Distinct Characters
Check if a String is Subsequence of Other
Class 2:  - LINKED LIST
  - Introduction to Linked List
  - Traversing a Linked List
  - Insertion of Node in Singly Linked List
  - Reverse a Linked List
  - Deletion of Node in Linked List
  - Doubly Linked List & Circular Linked List

Practice Problems
Middle of Linked List
Deleting a node without accessing head pointer of Linked List
Nth Node from end of Linked List
Segregating Even and Odd Nodes of LL
Detect Loop using Floyd Cycle Detection
LRU Cache

Week 05

Class 1:  - STACK
  - Stack - Introduction and Applications
  - Stack Operations (e.g. push, pop, etc)
  - Array Implementation of Stack
  - Linked List Implementation of Stack

Practice Problems
Balanced Parenthesis
Next Greater Element
Previous Greater Element
Implement two Stacks in an Array

Class 2:  - QUEUE
  - DEQUE
  - Queue- Introduction and Application
  - Implementation of Queue using Array
  - Implementation of Queue using Linked List
  - Deque - Introduction and Application
**Practice Problems**
Generate numbers with given digits
First Circular Tour
Maximums of all subarrays of size k
Reversing a Queue

---

**Class 1:**  
- **TREE**  
  - Tree - Introduction and Application  
  - Binary Tree  
  - Tree Traversal - Inorder, Preorder and Postorder with Implementation  
  - Level Order Traversal  
  - Set and Map  
  - TreeSet and TreeMap

**Practice Problems**
Height of Binary Tree
Maximum in a Binary Tree
Check for Balanced Binary Tree
Diameter of a Binary Tree
LCA of a Binary Tree
Serialize and Deserialize a Binary Tree

---

**Class 2:**  
- **BINARY SEARCH TREE**  
  - BST - Introduction and Application  
  - Search in BST with Implementation  
  - Insert in BST with Implementation  
  - Deletion in BST with Implementation  
  - Self Balancing BST - AVL Tree, Red Black Tree

**Practice Problems**
Find Kth Smallest in BST
Check for BST
Top View of Binary Tree
Vertical Sum in Binary Tree
Floor in BST
CONTENTS

Week 07

Class 1:  - **GREEDY**  
        - **HEAP**  
          - Introduction to Greedy Algorithm  
          - Binary Heap - Introduction  
          - Binary Heap - Insertion, Heapify and Extract  
          - Binary Heap - Decrease, Delete and Build Heap  
          - Heap Sort  
          - Priority Queue  

  **Practice Problems**  
  Activity Selection Problem  
  Job Sequencing Problem  
  Fractional Knapsack Problem  
  Sort K Sorted Array  
  K Largest Element  
  Median of a Stream

Class 2:  - **GRAPH**  
          - Introduction to Graph  
          - Graph Representation (Adjacency List and Matrix)  
          - Adjacency Matrix and List Comparison  
          - Breadth First Search - Introduction and Implementation  
          - Depth First Search - Introduction and Implementation  

  **Practice Problems**  
  Shortest Path in an Unweighted Graph  
  Detect Cycle in Undirected Graph  
  Detect Cycle in a Directed Graph  
  Strongly Connected Components

Week 08

Class 1:  - **Graph - Advanced**  
          - Prims Algorithm - Introduction and Implementation  
          - Dijkstra Algorithm - Introduction and Implementation  
          - Kruskal's Algorithm  
          - Bellman-Ford Algorithm
Practice Problems
Find the no. of islands
Articulation Point
Bridges in Graph

Class 2:
- Dynamic Programming
  - Introduction to Dynamic Programming
  - DP vs Greedy Approach
  - How to approach a DP Problem
  - Memoization and Tabulation methods

Practice Problems
Coin Change Problem
Longest Common Subsequence
Subset Sum Problem
Longest Increasing Subsequence
0-1 Knapsack Problem