Data Structures & Algorithm
(Basic to Intermediate)

- **Analysis of Algorithms:**
    - Growth of functions
    - Asymptotic Notations Omega, Theta,
    - Recursion Tree Method
    - Space Complexity
- **Arrays:**
    - Insertion, Deletion, Updation, Shifting
    - Reversal, Sort Check, Maximum, Minimum
- **Recursion**
    - Introduction to Recursion
    - Tail Recursion
    - Natural Number Check Using Recursion
    - Palindrome Check Using Recursion
    - Sum of Digits, Rod Cutting and Subsets
    - Tower of Hanoi
- **Hashing:**
    - Introduction to Hashing
    - Direct Address Table
    - Collision Handling
    - Chaining
    - Open Addressing
    - Double Hashing
    - Chaining Vs Open Addressing
- **String:**
    - Introduction to Strings
- **Searching:**
    - Linear Search
    - Binary Search (Iterative and Recursive)
- **Sorting:**
    - Stability in Sorting Algorithm
    - Bubble Sort
    - Selection Sort
    - Insertion Sort
    - Quick Sort
    - Different Partition Schemes in QuickSort
    - Merge Sort

    - Lomuto Partition
    - Hoare Partition
    - Heap Sort
    - Counting Sort
    - Radix Sort
    - Bucket Sort
- **Linked List:**
    - Drawback of Arrays
    - Introduction to Linked List and Implementation
    - Traversal, Insertion and Deletion
    - Sorted Insertion in Linked List
    - Reversal of Linked List (Iterative and Recursive)
    - Finding Middle
    - Remove Duplicate from Sorted Linked List
- **Circular Linked List:**
    - Traversal
    - Insertion (Head, End)
    - Deletion (Head, Kth Node)
- **Doubly Linked List:**
    - Traversal
    - Insertion (Head, End)
    - Deletion (Head, End)
    - Reversal
    - Circular Doubly Linked List
- **Stack:**
    - Introduction to Stack Data Structure
    - Implement using array
    - Implementation using Linked List
    - Stack Applications
- **Queue:**
    - Introduction to Queue Data Structure
    - Implementation using array
    - Implementation using Linked List.
- **Dequeue:**
    - Introduction to Deque Data Structure.
    - Implementations using Array
    - Implementation using Linked List
- **Tree:**
    - Implementation
    - Traversals: preorder, postorder, inorder, level order(Iterative & Recursive)
    - Binary Tree: Height, Size, Maximum
    - Print Nodes at K Distance

- **BST:**
    - Implementation
    - Search
    - Insertion
    - Deletion
    - Floor and Ceil in BST in CPP and Java
    - Self Balancing BST
    - AVL Tree (Introduction and applications)
    - Red-Black Tree (Introduction and applications)
    - Applications of BST
- **Heap:**
    - Implementation
    - Insert
    - Heapify and Extract in Heap
    - Decrease Key, Delete and Build Heap