Week 1 - Array Problems -------------------------------------------------------------------------

* Left Rotation of the array by 1
* Left Rotation of the array by D places
* Leaders in an Array
* Maximum Difference Problem
* Stock Buy and Sell Problem
* Trapping Rainwater Problem
* Maximum subarray sum
* Longest even-odd subarray
* Maximum Circular sum subarray.
* Majority Element
* Sliding Window Technique
* Prefix sum technique etc.

Week 2 - **Bit Manipulation Problems----------------------**

Check Kth bit is set or not

Count Set Bits

* 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.

**Hashing-------------------------------------------------------------**

* 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

Week 3 & 4 - **Strings** ------------------------------------------------------

* 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.

**------------------------------------------Linked Lists --------------------------------------------------------**

* 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
* 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

**------------------------------------------Stacks --------------------------------------------------------**

* 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

**------------------------------------------Queue --------------------------------------------------------**

* Reversing a Queue
* Generate numbers with given digits
* Maximums of all subarrays of size k

Week 5 - **Binary Tree -----------------------------------------------------------------------------**

* 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
* The diameter of a Binary Tree
* LCA problem with an efficient solution

**------------------------------------------Binary Search Tree -----------------------------------------------------**

* 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

Week 6 – **Heaps ---------------------------------------------------------------------------------------------------**

* Sort K-Sorted Array
* Buy Maximum Items with Given Sum
* K Largest Elements
* Merge K Sorted Arrays
* Median of a Stream

**------------------------------------------Graphs ------------------------------------------------**

* + Shortest Path in an Unweighted Graph
  + Number of Islands
  + Snake-Ladder
  + 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

**------------------------------------------Greedy Algorithm ------------------------------------------------**

* Activity Selection Problem
* Fractional Knapsack
* Job Sequencing Problem

**-----------------------------------------Dynamic Programming--------------------------------------------**

* 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