**Arrays**

1) Set Matrix Zeroes

2) Pascal’s Triangle

3) Next Permutation

4) Kadane’s Algorithm

5) Sort an array of 0’s 1’s 2’s

6) Stock buy and Sell

**Arrays Part-II**

1) Rotate Matrix

2) Merge Overlapping Subintervals

3) Merge two sorted Arrays without extra space

4) Find the duplicate in an array of N+1 integers.

5) Repeat and Missing Number

6) Inversion of Array (Pre-req: Merge Sort)

**Arrays Part-III**

1) Search in a 2d Matrix

2) Pow(X,n)

3) Majority Element (>N/2 times)

4) Majority Element (>N/3 times)

5) Grid Unique Paths

6) Reverse Pairs (Leetcode)

**Arrays Part-IV**

1) 2-Sum-Problem

2) 4-sum-Problem

3) Longest Consecutive Sequence

4) Largest Subarray with 0 sum

5) Count number of subarrays with given Xor K

6) Longest Substring without repeat

**Linked List**

1) Reverse a LinkedList

2) Find the middle of LinkedList

3) Merge two sorted Linked List (use method used in mergeSort)

4) Remove N-th node from back of LinkedList

5) Add two numbers as LinkedList

6) Delete a given Node when a node is given.(0(1) solution)

**Linked List Part-II**

1) Find intersection point of Y LinkedList

2) Detect a cycle in Linked List

3) Reverse a LinkedList in groups of size k.

4) Check if a LinkedList is palindrome or not.

5) Find the starting point of the Loop of LinkedList

6) Flattening of a LinkedList

**Linked List and Arrays**

1) Rotate a LinkedList

2) Clone a Linked List with random and next pointer

3) 3 sum

4) Trapping rainwater

5) Remove Duplicate from Sorted array

6) Max consecutive ones

**Greedy Algorithm**

1) N meetings in one room

2) Minimum number of platforms required for a railway

3) Job sequencing Problem

4) Fractional Knapsack Problem

5) Greedy algorithm to find minimum number of coins

6) Activity Selection (it is the same as N meeting in one room)

**Recursion**

1) Subset Sums

2) Subset-II

3) Combination sum-1

4) Combination sum-2

5) Palindrome Partitioning

6) K-th permutation Sequence

**Recursion and Backtracking**

1) Print all permutations of a string/array

2) N queens Problem

3) Sudoku Solver

4) M coloring Problem

5) Rat in a Maze

6) Word Break (print all ways)

**Binary Search**

1) The N-th root of an integer

2) Matrix Median

3) Find the element that appears once in a sorted array, and the rest element appears twice (Binary search)

4) Search element in a sorted and rotated array/ find pivot where it is rotated

5) Median of 2 sorted arrays

6) K-th element of two sorted arrays

7) Allocate Minimum Number of Pages

8) Aggressive Cows

**Heaps**

1) Max heap, Min Heap Implementation (Only for interviews)

2) Kth Largest Element

3) Maximum Sum Combination

4) Find Median from Data Stream

5) Merge K sorted arrays

6) K most frequent elements

**Stack and Queue**

1) Implement Stack Using Arrays

2) Implement Queue Using Arrays

3) Implement Stack using Queue (using single queue)

4) Implement Queue using Stack (0(1) amortized method)

5) Check for balanced parentheses

6) Next Greater Element

7) Sort a Stack

**Stack and Queue Part-II**

1) Next Smaller Element

2) LRU cache (IMPORTANT)

3) LFU Cache

4) Largest rectangle in a histogram

5) Sliding Window maximum

6) Implement Min Stack

7) Rotten Orange (Using BFS)

8) Stock Span Problem

9) Find the maximum of minimums of every window size

10) The Celebrity Problem

**String**

1) Reverse Words in a String

2) Longest Palindrome in a string

3) Roman Number to Integer and vice versa

4) Implement ATOI/STRSTR

5) Longest Common Prefix

6) Rabin Karp

**String Part-II**

1) Z-Function

2) KMP algo / LPS(pi) array

3) Minimum characters needed to be inserted in the beginning to make it palindromic

4) Check for Anagrams

5) Count and Say

6) Compare version numbers

**Binary Tree**

1) Inorder Traversal

2) Preorder Traversal

3) Postorder Traversal

4) Morris Inorder Traversal

5) Morris Preorder Traversal

6) LeftView Of Binary Tree

7) Bottom View of Binary Tree

8) Top View of Binary Tree

9) Preorder inorder postorder in a single traversal

10) Vertical order traversal

11) Root to node path in a Binary Tree

12) Max width of a Binary Tree

**Binary Tree part-II**

1) Level order Traversal / Level order traversal in spiral form

2) Height of a Binary Tree

3) Diameter of Binary Tree

4) Check if the Binary tree is height-balanced or not

5) LCA in Binary Tree

6) Check if two trees are identical or not

7) Zig Zag Traversal of Binary Tree

8) Boundary Traversal of Binary Tree

**Binary Tree part-III**

1) Maximum path sum

2) Construct Binary Tree from inorder and preorder

3) Construct Binary Tree from Inorder and Postorder

4) Symmetric Binary Tree

5) Flatten Binary Tree to LinkedList

6) Check if Binary Tree is the mirror of itself or not

7) Check for Children Sum Property

**Binary Search Tree**

1) Populate Next Right pointers of Tree

2) Search given Key in BST

3) Construct BST from given keys

4) Construct BST from preorder traversal

5) Check is a BT is BST or not

6) Find LCA of two nodes in BST

7) Find the inorder predecessor/successor of a given Key in BST.

**Binary Search Tree Part-II**

1) Floor in a BST

2) Ceil in a BST

3) Find K-th smallest element in BST

4) Find K-th largest element in BST

5) Find a pair with a given sum in BST

6) BST iterator

7) Size of the largest BST in a Binary Tree

8) Serialize and deserialize Binary Tree

**Binary Trees[Miscellaneous]**

1) Binary Tree to Double Linked List

2) Find median in a stream of running integers.

3) K-th largest element in a stream.

4) Distinct numbers in Window.

5) K-th largest element in an unsorted array.

6) Flood-fill Algorithm

**Graph**

1) Clone a graph (Not that easy as it looks)

2) DFS

3) BFS

4) Detect A cycle in Undirected Graph using BFS

5) Detect A cycle in Undirected Graph using DFS

6) Detect A cycle in a Directed Graph using DFS

7) Detect A cycle in a Directed Graph using BFS

8) Topological Sort BFS

9) Topological Sort DFS

10) Number of islands(Do in Grid and Graph Both)

11) Bipartite Check using BFS

12) Bipartite Check using DFS

**Graph Part-II**

1) Strongly Connected Component(using KosaRaju’s algo)

2) Dijkstra’s Algorithm

3) Bellman-Ford Algo

4) Floyd Warshall Algorithm

5) MST using Prim’s Algo

6) MST using Kruskal’s Algo

**Dynamic Programming**

1) Max Product Subarray

2) Longest Increasing Subsequence

3) Longest Common Subsequence

4) 0-1 Knapsack

5) Edit Distance

6) Maximum sum increasing subsequence

7) Matrix Chain Multiplication

**Dynamic Programming Part-II**

1) Minimum sum path in the matrix, (count paths and similar type do, also backtrack to find the Minimum path)

2) Coin change

3) Subset Sum

4) Rod Cutting

5) Egg Dropping

6) Word Break

7) Palindrome Partitioning (MCM Variation)

8) Maximum profit in Job scheduling

**Trie**

1) Implement Trie (Prefix Tree)

2) Implement Trie – 2 (Prefix Tree)

3) Longest String with All Prefixes

4) Number of Distinct Substrings in a String

5) Power Set (this is very important)

6) Maximum XOR of two numbers in an array

7) Maximum XOR With an Element From Array