Arrays

1. Set Matrix Zeros
2. **Pascal Triangle**
3. Next Permutation (Was Asked)
4. Kadane’s Algorithm
5. **Max Product Subarray**
6. Sort an array of 0’s 1’s 2’s
7. Stock Buy and Sell (Asked)
8. Rotate Matrix
9. **Merge Overlapping Subintervals**
10. Merge 2 arrays without extra space (Leetcode)
11. Merge 2 arrays without extra space (GFG)
12. **Find the duplicate in an array of N+1 integers.**
13. **Repeat and Missing Number**
14. **Inversion of Array (Pre-req: Merge Sort)**
15. Search in a 2d Matrix (Leetcode)
16. **Search in a 2d Matrix (GFG)**
17. **Pow(X,n)**
18. Majority Element (>N/2 times) **(Asked)**
19. **Majority Element (>N/3 times)**
20. **Grid Unique Paths (Google Interview Question)**
21. **Reverse Pairs (Leetcode)**
22. 2-Sum-Problem
23. **4-Sum-Problem**
24. 3-Sum-Problem
25. **Longest Consecutive Sequence** (Asked)
26. Check if subarray with 0 sum exists or not
27. **Largest Subarray with 0 sum**
28. **Count number of subarrays with given XOR K**
29. **Longest Substring without repeat** (Asked)
30. **Trapping rainwater (Space O(1))**
31. Remove Duplicate from Sorted array
32. Max consecutive ones
33. **Spiral Matrix**

Linked List

1. **Reverse a LinkedList** (Asked)
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 (One pass approach needs to be remembered)**
5. **Add two numbers as LinkedList (Strivers code if efficient)** (Asked)
6. Delete a given Node when a node is given. (0(1) solution)
7. **Find intersection point of Y LinkedList (1 normal + 2 optimal approaches)**
8. Detect a cycle in Linked List
9. **Find the starting point of the Loop of LinkedList**
10. **Reverse a LinkedList in groups of size k. (Asked)**
11. **Check if a LinkedList is palindrome or not.**
12. Flattening of a LinkedList (Asked)
13. Rotate a LinkedList
14. **Clone a Linked List with random and next pointer**

Greedy

1. N meeting in one room + Activity Selection (it is the same as N meeting 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. A edge case q of bigger DP Problem!
6. Dijkstra’s Algorithm. (Part of graphs, but it makes me realize that Dijkstra’s is easy)

Recursion & Backtracking

1. Subset Sums
2. Subset 2
3. Combination sum-1
4. Combination sum-2
5. Palindrome Partitioning
6. K-th permutation Sequence
7. Print all permutations of a string/array
8. **N queens Problem**
9. Sudoku Solver
10. **M coloring Problem**
11. **Rat in a Maze**
12. Word Break (print all ways)

Binary Search

1. **Binary Search Aditya Verma Playlist**
2. **The N-th root of an integer**
3. Matrix Median
4. **Find the element that appears once in a sorted array, and the rest element appears twice (Binary search)**
5. **Search element in a sorted and rotated array/ find pivot where it is rotated**
6. **Median of 2 sorted arrays**
7. **K-th element of two sorted arrays**
8. **Allocate Minimum Number of Pages**
9. **Aggressive Cows**
10. **Painters Partition (Copy of Book allocation)**

Tries (Asked theoretically)

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

Stack & 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**
8. Next Smaller Element
9. **LRU cache (IMPORTANT)**
10. LFU Cache
11. **Largest rectangle in a histogram**
12. **Sliding Window maximum**
13. **Implement Min Stack**
14. **Rotten Orange (Using BFS)**
15. **Stock Span Problem**
16. Find the maximum of minimums of every window size (too tough)
17. **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 (done)
7. Z-Function
8. KMP algo / LPS(pi) array
9. Minimum characters needed to be inserted in the beginning to make it palindromic
10. Check for Anagrams (Done)
11. Count and Say
12. Compare version numbers

Binary Tree

1. Inorder Traversal (Iteratively)
2. Preorder Traversal (iteratively)
3. **Postorder Traversal (iteratively)**
4. **Preorder inorder postorder in a single traversal**
5. Level order Traversal
6. **Level order traversal in spiral form** **or Zig Zag Traversal of Binary Tree (Imp to solve it using Striver approach)**
7. Height of a Binary Tree
8. **Left View Of Binary Tree**
9. Right View of Binary Tree
10. **Vertical order traversal**
11. Bottom View of Binary Tree
12. Top View of Binary Tree
13. Boundary Traversal of Binary Tree
14. Check if the Binary tree is height-balanced or not **(asked)**
15. Diameter of Binary Tree
16. **Check if two trees are identical or not (retry)**
17. **Symmetric Binary Tree (retry)**
18. Root to node path in a Binary Tree
19. Invert a binary tree
20. **Construct Binary Tree from inorder and preorder**
21. Construct Binary Tree from Inorder and Postorder
22. **Check for Children Sum Property**
23. **Max width of a Binary Tree**
24. **Maximum path sum**
25. **LCA in Binary Tree (asked)**
26. Find LCA of two nodes in BST **(asked)**
27. **Morris Inorder Traversal**
28. Morris Preorder Traversal
29. **Flatten Binary Tree to LinkedList** **(asked)**
30. **Connect nodes at same level** **(asked) (with extra space and without)**
31. Binary Tree to Double Linked List
32. Cousins in Binary Tree **(asked)**
33. Sum of leaf nodes at min level **(asked)**
34. **Nodes at K distance from given node (Imp concept)**
35. Minimum time to burn the binary tree

BST

1. Search given Key in BST
2. Floor in a BST
3. Ceil in a BST
4. Find K-th smallest element in BST
5. Find K-th largest element in BST
6. **BST Iterator**
7. Find a pair with a given sum in BST
8. **Check is a BT is BST or not (asked)**
9. **Size of the largest BST in a Binary Tree (Tough)**
10. Construct BST from given inorder traversal
11. **Construct BST from preorder traversal**
12. **Find the inorder predecessor/successor of a given Key in BST. (asked)**
14. Serialize and deserialize Binary Tree

Graphs

1. **Clone a graph**
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**
13. Strongly Connected Component(using KosaRaju’s algo)
14. **Dijkstra’s Algorithm**
15. **Bellman-Ford Algo**
16. Floyd Warshall Algorithm
17. **MST using Prim’s Algo (don’t like this)**
18. **MST using Kruskal’s Algo (again, not a big fan)**
19. Word Ladder
20. M coloring Problem
21. Flood-fill Algorithm **(easy)**

Priority Queue(Heap)

1. Find median in a stream of running integers.
2. K-th largest element in a stream
3. K-th largest element in an unsorted array

Two Pointer/Sliding Window

1. Distinct numbers in Window.**(asked)**

DP