Data Structure specific algorithms

1. Arrays

- Sorting:
  - QuickSort: Efficient average-case time complexity (O(nlog n))
  - MergeSort: Stable sort, useful when order matters (O(nlog n))
- Searching:
  - Binary Search: Fast search in sorted arrays (O(log n))
- Two Pointers:
  - In-place manipulation, often for sorted arrays (e.g., removing duplicates)
- Sliding Window:
  - Subarray problems, finding maximum/minimum within a window

2. Linked Lists

- Traversal:
  - Iterate through the list, understand the node structure
- Insertion/Deletion:
  - At beginning, end, or at a specific position
- Reversal:
  - In-place reversal, recursive and iterative approaches
- Cycle Detection:
  - Floyd's Tortoise and Hare algorithm

3. Hash Tables (Hash Maps/Sets)

- Implementation not needed. Just understand following:
  - Understand how hash functions work
  - Insertion/Deletion/Lookup
  - Collision Handling

4. Trees (Binary Trees, Binary Search Trees, etc.)

- Traversal:
  - Inorder, Preorder, Postorder (recursive and iterative)
- Searching:
  - Find a node with a given value (especially in BSTs)

5. Stacks

- Implementation not needed. Just understand following:
  - Push/Pop/Peek Operations

6. Queues

- Implementation not needed. Just understand following:
  - Enqueue/Dequeue Operations

7. Heaps (Priority Queues)

- Implementation not needed. Just understand following:
  - Insertion/Deletion (extract-min/max)
  - Building a Heap
- Top K Elements:
  - Using a heap to find k largest/smallest elements

8. Graphs

- Traversal:
  - Breadth-First Search (BFS)
  - Depth-First Search (DFS)
- Shortest Path:
  - Dijkstra's Algorithm
- Cycle Detection:
  - DFS

9. Tries

- Implement Trie from scratch
- Insertion/Searching:

        - For words/prefixes
- Autocompletion:
        - Using a trie for word suggestions

10. Union-Find (Disjoint Set)

- Implement Union-Find from scratch
- Find/Union Operations
- Cycle Detection in undirected graphs

General algorithms/techniques

1. Recursion

- Defining a problem in terms of itself, often leading to elegant and concise solutions.
- Solve: Factorial calculation, tree traversals, depth-first search.

2. Dynamic Programming

- Breaking down a problem into overlapping subproblems and storing solutions to avoid recomputation.
- Solve: Fibonacci sequence, Knapsack problem, Longest Common Subsequence.

3. Greedy Algorithms

- Making locally optimal choices at each step with the hope of finding a global optimum.
- Implement: Kruskal's algorithm for minimum spanning trees.

4. Backtracking

- Incrementally building solutions, exploring all possible paths, and abandoning invalid ones.
- Solve: Sudoku solver, N-Queens problem, generating permutations.

Interview Master 100

1. Two Sum [Solution]
2. Valid Parentheses
3. Merge Two Sorted Lists [Solution]
4. Best Time to Buy and Sell Stock [Solution]
5. Valid Palindrome [Solution]
6. Invert Binary Tree
7. Valid Anagram
8. Binary Search
9. Linked List Cycle
10. Maximum Depth of Binary Tree
11. Single Number
12. Reverse Linked List
13. Majority Element
14. Missing Number
15. Reverse String
16. Diameter of Binary Tree
17. Middle of the Linked List
18. Convert Sorted Array to Binary Search Tree
19. Maximum Subarray [Solution]
20. Climbing Stairs [Solution]
21. Symmetric Tree [Solution]
22. Product of Array Except Self [Solution]
23. Best Time to Buy and Sell Stock II
24. House Robber [Solution]
25. Number of 1 Bits
26. Validate Binary Search Tree
27. Min Stack [Solution]
28. Contains Duplicate [Solution]
29. Kth Smallest Element in a BST

30. Merge Intervals [Solution]
31. Set Matrix Zeroes [Solution]
32. Spiral Matrix [Solution]
33. 3Sum [Solution]
34. Binary Tree Zigzag Level Order Traversal
35. Construct Binary Tree from Preorder and Inorder Traversal
36. Container With Most Water [Solution]
37. Flatten Binary Tree to Linked List
38. Group Anagrams
39. Implement Trie (Prefix Tree)
40. Kth Largest Element in an Array
41. Longest Palindromic Substring
42. Longest Substring Without Repeating Characters [Solution]
43. Maximal Square [Solution]
44. Maximum Product Subarray
45. Minimum Window Substring
46. Number of Islands [Solution]
47. Permutations
48. Remove Nth Node From End of List
49. Rotate Image [Solution]
50. Search a 2D Matrix
51. Search in Rotated Sorted Array
52. Subsets [Solution]
53. Top K Frequent Elements [Solution]
54. Trapping Rain Water
55. Two Sum II - Input Array Is Sorted
56. Unique Paths
57. Valid Sudoku
58. Word Break
59. Word Search
60. Add Two Numbers [Solution]
61. Basic Calculator
62. Coin Change
63. Combination Sum
64. Copy List with Random Pointer
65. Course Schedule
66. Design Add and Search Words Data Structure
67. Merge Sorted Array
68. Find Median from Data Stream
69. Game of Life
70. Jump Game
71. Letter Combinations of a Phone Number
72. Longest Consecutive Sequence [Solution]
73. Longest Increasing Subsequence
74. Median of Two Sorted Arrays
75. Merge k Sorted Lists [Solution]
76. Minimum Path Sum
77. Word Search II
78. Reverse Nodes in k-Group
79. Course Schedule II
80. Remove Element
81. Rotate Array
82. Bitwise AND of Numbers Range

83. Palindrome Number
84. Plus One
85. Sqrt(x)
86. Pow(x n) [Solution]
87. Construct Binary Tree from Inorder and Postorder Traversal
88. Path Sum
89. Binary Tree Right Side View
90. Binary Tree Level Order Traversal [Solution]
91. Minimum Absolute Difference in BST
92. Surrounded Regions
93. Clone Graph
94. Evaluate Division
95. Generate Parentheses [Solution]
96. Sort List
97. Maximum Sum Circular Subarray
98. Find Peak Element
99. Find Minimum in Rotated Sorted Array [Solution]
100. Remove Duplicates from Sorted Array