100 Problem-Solving Questions in Data Structures and Algorithms (DSA)

1. Arrays

1. Find the second largest element in an array.
2. Reverse an array.
3. Find the missing number in a given integer array of 1 to N.
4. Find the duplicate number in an array containing N+1 integers.
5. Merge two sorted arrays without extra space.
6. Find the intersection and union of two arrays.
7. Rotate an array by K places.
8. Find the subarray with the maximum sum (Kadane's Algorithm).
9. Move all zeroes to the end of the array.
10. Find the longest consecutive sequence in an array.

2. Strings

11. Reverse a string.
12. Check if a string is a palindrome.
13. Find the first non-repeating character in a string.
14. Check if two strings are anagrams.
15. Convert a string to an integer (atoi).
16. Find the longest common prefix among a set of strings.
17. Implement the Rabin-Karp algorithm for pattern searching.
18. Implement the KMP algorithm for pattern searching.
19. Find the longest repeating subsequence in a string.
20. Count and say problem.

3. Linked Lists

21. Reverse a linked list.
22. Detect and remove a loop in a linked list.
23. Find the middle element of a linked list.
24. Merge two sorted linked lists.
25. Find the intersection point of two linked lists.
26. Remove duplicates from a sorted linked list.
27. Check if a linked list is a palindrome.

28. Add two numbers represented by linked lists.
29. Flatten a multilevel linked list.
30. Rotate a linked list.

4. Stacks and Queues

31. Implement a stack using arrays.
32. Implement a queue using arrays.
33. Implement a stack using queues.
34. Implement a queue using stacks.
35. Implement the LRU cache.
36. Check for balanced parentheses in an expression.
37. Evaluate a postfix expression.
38. Implement a circular queue.
39. Implement a Min Stack.
40. Implement the Next Greater Element algorithm.

5. Recursion and Backtracking

41. Find all permutations of a string.
42. Find all subsets of a set.
43. Solve the N-Queens problem.
44. Solve the Rat in a Maze problem.
45. Generate balanced parentheses.
46. Implement the Tower of Hanoi.
47. Find the kth permutation sequence.
48. Word break problem.
49. Print all possible paths in a matrix.
50. Find the shortest path in a maze.

6. Binary Trees

51. Preorder, Inorder, and Postorder traversals of a binary tree.
52. Level order traversal of a binary tree.
53. Check if a binary tree is height-balanced.
54. Find the lowest common ancestor of two nodes.
55. Convert a binary tree to a linked list.

56. Check if two binary trees are identical.
57. Count the number of nodes in a binary tree.
58. Find the diameter of a binary tree.
59. Mirror a binary tree.
60. Find the sum of nodes at the Kth level.

7. Binary Search Trees (BSTs)

61. Insert a node in a BST.
62. Delete a node in a BST.
63. Find the Kth smallest element in a BST.
64. Check if a binary tree is a BST.
65. Convert a BST to a balanced BST.
66. Find the inorder successor of a node.
67. Find the lowest common ancestor in a BST.
68. Convert a BST to a greater sum tree.
69. Find pairs in a BST that sum to K.
70. Implement a BST iterator.

8. Graphs

71. Implement BFS for a graph.
72. Implement DFS for a graph.
73. Find the shortest path in an unweighted graph.
74. Detect a cycle in an undirected graph.
75. Detect a cycle in a directed graph.
76. Find the topological sort of a graph.
77. Implement Dijkstra's algorithm.
78. Implement Floyd-Warshall algorithm.
79. Implement Prim's algorithm for Minimum Spanning Tree.
80. Implement Kruskal's algorithm for Minimum Spanning Tree.

9. Dynamic Programming

81. Find the nth Fibonacci number (Top-down and Bottom-up).
82. Solve the 0/1 Knapsack problem.
83. Find the longest common subsequence.

84. Solve the subset sum problem.
85. Find the minimum number of insertions to make a string palindrome.
86. Solve the coin change problem.
87. Find the longest increasing subsequence.
88. Solve the edit distance problem.
89. Find the maximum sum increasing subsequence.
90. Solve the rod cutting problem.

10. Miscellaneous

91. Implement a trie (prefix tree).
92. Implement an LFU cache.
93. Find the median in a data stream.
94. Implement a disjoint set union-find.
95. Solve the gas station problem.
96. Solve the largest rectangle in a histogram problem.
97. Find the maximum number of meetings in a room.
98. Solve the word ladder problem.
99. Find the minimum spanning tree using Prim's algorithm.
100. Implement a sliding window maximum problem.

This collection of problems will help you strengthen your problem-solving skills in DSA. Practice them to become proficient in coding interviews!