LeetCode 75 Problems - Data Structure & Algorithm Classification

Array & Hashing (9 problems)

1. **Two Sum** - Hash table lookup
2. **Best Time to Buy and Sell Stock** - Single pass tracking
3. **Contains Duplicate** - Hash set for duplicates
4. **Product of Array Except Self** - Prefix/suffix products
5. **Maximum Subarray** - Kadane's algorithm
6. **Maximum Product Subarray** - Dynamic programming variant
7. **Find Minimum in Rotated Sorted Array** - Modified binary search
8. **Search in Rotated Sorted Array** - Modified binary search
9. **3Sum** - Two pointers with sorting

Two Pointers (4 problems)

10. **Container With Most Water** - Two pointers optimization
11. **Valid Palindrome** - String validation with two pointers
12. **Two Sum II - Input Array Is Sorted** - Two pointers on sorted array
13. **3Sum** - (Also uses two pointers technique)

Sliding Window (4 problems)

14. **Longest Substring Without Repeating Characters** - Variable window
15. **Longest Repeating Character Replacement** - Variable window with frequency
16. **Minimum Window Substring** - Variable window matching
17. **Best Time to Buy and Sell Stock** - (Can also use sliding window approach)

Stack (3 problems)

18. **Valid Parentheses** - Stack for matching brackets
19. **Longest Valid Parentheses** - Stack with indices
20. **Largest Rectangle in Histogram** - Stack for area calculation

Binary Search (3 problems)

21. **Find Minimum in Rotated Sorted Array** - Modified binary search
22. **Search in Rotated Sorted Array** - Modified binary search

23. **Search a 2D Matrix** - 2D binary search

Linked List (6 problems)

24. **Reverse Linked List** - Iterative/recursive reversal
25. **Detect Cycle in a Linked List** - Floyd's cycle detection
26. **Merge Two Sorted Lists** - Two pointer merging
27. **Merge k Sorted Lists** - Divide and conquer/heap
28. **Remove Nth Node From End of List** - Two pointers
29. **Reorder List** - Find middle + reverse + merge

Trees (12 problems)

30. **Maximum Depth of Binary Tree** - DFS/BFS traversal
31. **Same Tree** - Tree comparison
32. **Invert/Flip Binary Tree** - Tree transformation
33. **Binary Tree Maximum Path Sum** - DFS with global max
34. **Binary Tree Level Order Traversal** - BFS traversal
35. **Serialize and Deserialize Binary Tree** - Tree encoding/decoding
36. **Subtree of Another Tree** - Tree matching
37. **Construct Binary Tree from Preorder and Inorder Traversal** - Tree construction
38. **Validate Binary Search Tree** - BST property validation
39. **Kth Smallest Element in a BST** - In-order traversal
40. **Lowest Common Ancestor of BST** - BST property utilization
41. **Implement Trie (Prefix Tree)** - Trie data structure

Heap/Priority Queue (3 problems)

42. **Merge k Sorted Lists** - Min heap for merging
43. **Top K Frequent Elements** - Heap/bucket sort
44. **Find Median from Data Stream** - Two heaps (max + min)

Backtracking (4 problems)

45. **Combination Sum** - Backtracking with duplicates allowed
46. **Word Search** - 2D grid backtracking
47. **Palindrome Partitioning** - String partitioning
48. **N-Queens** - Classic backtracking problem

Tries (2 problems)

49. **Implement Trie (Prefix Tree)** - Basic trie operations
50. **Word Search II** - Trie with DFS

Graphs (7 problems)

51. **Number of Islands** - DFS/BFS on 2D grid
52. **Clone Graph** - Graph traversal and copying
53. **Pacific Atlantic Water Flow** - DFS from boundaries
54. **Course Schedule** - Topological sort/cycle detection
55. **Course Schedule II** - Topological sort with ordering
56. **Graph Valid Tree** - Cycle detection + connectivity
57. **Number of Connected Components in Undirected Graph** - Union-find/DFS

Advanced Graphs (3 problems)

58. **Alien Dictionary** - Topological sort on characters
59. **Word Ladder** - BFS shortest path
60. **Longest Consecutive Sequence** - Union-find or hash set

Dynamic Programming (11 problems)

61. **Climbing Stairs** - Basic DP (Fibonacci-like)
62. **Coin Change** - Unbounded knapsack
63. **Longest Increasing Subsequence** - Classic LIS
64. **Longest Common Subsequence** - 2D DP
65. **Word Break Problem** - String DP
66. **Combination Sum IV** - DP counting combinations
67. **House Robber** - Linear DP
68. **House Robber II** - Circular array DP
69. **Decode Ways** - String decoding DP
70. **Unique Paths** - Grid path counting
71. **Jump Game** - Greedy/DP reachability

Greedy (2 problems)

72. **Jump Game** - Greedy reachability check
73. **Maximum Subarray** - Kadane's algorithm (greedy approach)

Intervals (4 problems)

74. **Insert Interval** - Interval manipulation
75. **Merge Intervals** - Interval merging
76. **Non-overlapping Intervals** - Greedy interval scheduling
77. **Meeting Rooms** - Interval overlap checking

Math & Bit Manipulation (4 problems)

78. **Reverse Integer** - Integer manipulation
79. **Number of 1 Bits** - Bit counting
80. **Counting Bits** - DP on bit patterns
81. **Missing Number** - XOR/arithmetic tricks

Summary by Category

Category	Count	Key Concepts
Trees	12	DFS, BFS, BST properties, tree construction
Dynamic Programming	11	Optimal substructure, memoization
Array & Hashing	9	Hash tables, prefix sums, Kadane's algorithm
Graphs	7	DFS, BFS, topological sort, cycle detection
Linked List	6	Two pointers, cycle detection, merging
Intervals	4	Sorting, greedy merging
Sliding Window	4	Variable/fixed window optimization
Two Pointers	4	Left-right pointers, sorted arrays
Backtracking	4	Recursive exploration, pruning
Math & Bit Manipulation	4	Bit operations, mathematical properties
Heap/Priority Queue	3	Min/max heaps, top-k problems
Advanced Graphs	3	Shortest path, topological sort
Stack	3	LIFO operations, matching problems
Binary Search	3	Divide and conquer, sorted arrays
Tries	2	Prefix trees, word problems
Greedy	2	Local optimal choices

Study Recommendations

Beginner Path: Start with Array & Hashing → Two Pointers → Sliding Window → Stack → Linked List

Intermediate Path: Trees → Dynamic Programming → Graphs → Binary Search

Advanced Path: Backtracking → Tries → Advanced Graphs → Intervals → Math & Bit Manipulation

Note: Some problems appear in multiple categories as they can be solved using different approaches or combine multiple techniques.