Chapter 1–2: Arrays & Binary Search

- [ ] 1. Two Sum (Easy) — Array basics, Hash Table
- [ ] 167. Two Sum II (Easy) — Sorted array, Two pointers
- [ ] 26. Remove Duplicates from Sorted Array (Easy) — Array manipulation
- [ ] 27. Remove Element (Easy) — Array deletion
- [ ] 35. Search Insert Position (Easy) — Binary Search basics
- [ ] 704. Binary Search (Easy) — Binary Search implementation
- [ ] 278. First Bad Version (Easy) — Binary Search variant
- [ ] 34. Find First and Last Position (Medium) — Binary Search bounds
- [ ] 33. Search in Rotated Sorted Array (Medium) — Modified Binary Search
- [ ] 153. Find Minimum in Rotated Array (Medium) — Binary Search application
- [ ] 4. Median of Two Sorted Arrays (Hard) — Advanced Binary Search

Chapter 3–5: Sorting Algorithms

- [ ] 912. Sort an Array (Medium) — Implement sorting algorithms
- [ ] 75. Sort Colors (Medium) — Three-way partitioning
- [ ] 242. Valid Anagram (Easy) — Sorting application
- [ ] 88. Merge Sorted Array (Easy) — Merging sorted arrays
- [ ] 148. Sort List (Medium) — Merge sort on linked list
- [ ] 56. Merge Intervals (Medium) — Sorting application

Chapter 6: Insertion Sort

- [ ] 147. Insertion Sort List (Medium) — Insertion sort implementation
- [ ] 708. Insert into a Sorted Circular Linked List (Medium) — Insertion practice

Chapter 7: Complexity Analysis

- [ ] 217. Contains Duplicate (Easy) — O(n) time complexity
- [ ] 219. Contains Duplicate II (Easy) — Sliding window
- [ ]

169. Majority Element (Easy) — Frequency counting

- [ ] 268. Missing Number (Easy) — Mathematical approach
- [ ] 448. Find All Numbers Disappeared (Easy) — Array marking

Chapter 8: Hash Tables

- [ ] 49. Group Anagrams (Medium) — Hash table grouping
- [ ] 128. Longest Consecutive Sequence (Medium) — Hash set operations
- [ ] 383. Ransom Note (Easy) — Hash map usage
- [ ] 387. First Unique Character (Easy) — Hash map ordering
- [ ] 290. Word Pattern (Easy) — Hash map mapping
- [ ] 347. Top K Frequent Elements (Medium) — Hash + Heap
- [ ] 349. Intersection of Two Arrays (Easy) — Set operations
- [ ] 350. Intersection of Two Arrays II (Easy) — Hash map counting

Chapter 9: Stacks & Queues

- [ ] 20. Valid Parentheses (Easy) — Stack basics
- [ ] 155. Min Stack (Medium) — Stack design
- [ ] 232. Implement Queue using Stacks (Easy) — Queue implementation
- [ ] 225. Implement Stack using Queues (Easy) — Stack implementation
- [ ] 150. Evaluate Reverse Polish (Medium) — Stack application
- [ ] 739. Daily Temperatures (Medium) — Monotonic stack
- [ ] 84. Largest Rectangle in Histogram (Hard) — Monotonic stack
- [ ] 394. Decode String (Medium) — Stack for nested structure
- [ ] 71. Simplify Path (Medium) — Stack for path

Chapter 10–11: Recursion Basics

- [ ] 509. Fibonacci Number (Easy) — Basic recursion
- [ ] 70. Climbing Stairs (Easy) — Recursive relation
- [ ] 344. Reverse String (Easy) — Simple recursion
- [ ]

206. Reverse Linked List (Easy) — Recursive list reversal

- [ ] 21. Merge Two Sorted Lists (Easy) — Recursive merging
- [ ] 24. Swap Nodes in Pairs (Medium) — Recursive swapping
- [ ] 50. Pow(x, n) (Medium) — Divide and conquer
- [ ] 779. K-th Symbol in Grammar (Medium) — Recursive pattern
- [ ] 22. Generate Parentheses (Medium) — Backtracking recursion
- [ ] 46. Permutations (Medium) — Backtracking
- [ ] 77. Combinations (Medium) — Backtracking
- [ ] 78. Subsets (Medium) — Backtracking subsets

Chapter 12: Dynamic Programming

- [ ] 198. House Robber (Medium) — 1D DP
- [ ] 213. House Robber II (Medium) — Circular DP
- [ ] 322. Coin Change (Medium) — Unbounded knapsack
- [ ] 518. Coin Change II (Medium) — Combination DP
- [ ] 91. Decode Ways (Medium) — String DP
- [ ] 62. Unique Paths (Medium) — Grid DP
- [ ] 63. Unique Paths II (Medium) — Grid DP with obstacles
- [ ] 55. Jump Game (Medium) — Greedy/DP
- [ ] 45. Jump Game II (Medium) — Greedy/DP optimization
- [ ] 139. Word Break (Medium) — String DP
- [ ] 300. Longest Increasing Subsequence (Medium) — Sequence DP
- [ ] 72. Edit Distance (Hard) — 2D DP
- [ ] 1143. Longest Common Subsequence (Medium) — 2D DP

Chapter 13: Quicksort & Quickselect

- [ ] 215. Kth Largest Element in an Array (Medium) — Quickselect classic
- [ ] 973. K Closest Points (Medium) — Quickselect application
- [ ] 324. Wiggle Sort II (Medium) — Partition + rearrange

- [ ] 347. Top K Frequent Elements (Medium) — Quickselect variant

Chapter 14: Linked Lists

- [ ] 141. Linked List Cycle (Easy) — Fast & slow pointers
- [ ] 142. Linked List Cycle II (Medium) — Cycle detection
- [ ] 160. Intersection of Two Lists (Easy) — Two pointer technique
- [ ] 234. Palindrome Linked List (Easy) — Multiple techniques
- [ ] 83. Remove Duplicates from Sorted List (Easy) — List traversal
- [ ] 203. Remove Linked List Elements (Easy) — Node deletion
- [ ] 2. Add Two Numbers (Medium) — List arithmetic
- [ ] 19. Remove Nth Node From End (Medium) — Two pointer
- [ ] 143. Reorder List (Medium) — Multiple operations
- [ ] 23. Merge K Sorted Lists (Hard) — Multiple list merging
- [ ] 25. Reverse Nodes in K-Group (Hard) — Complex reversal

Chapter 15: Binary Search Trees

- [ ] 700. Search in a BST (Easy) — BST search
- [ ] 701. Insert into a BST (Medium) — BST insertion
- [ ] 450. Delete Node in a BST (Medium) — BST deletion
- [ ] 98. Validate Binary Search Tree (Medium) — BST validation
- [ ] 94. Binary Tree Inorder Traversal (Easy) — BST traversal
- [ ] 144. Binary Tree Preorder Traversal (Easy) — Tree traversal
- [ ] 145. Binary Tree Postorder Traversal (Easy) — Tree traversal
- [ ] 102. Binary Tree Level Order (Medium) — BFS traversal
- [ ] 230. Kth Smallest in BST (Medium) — BST property
- [ ] 235. Lowest Common Ancestor BST (Medium) — BST ancestor
- [ ] 108. Convert Sorted Array to BST (Easy) — BST construction
- [ ] 110. Balanced Binary Tree (Easy) — Tree property
- [ ]

- [ ] 104. Maximum Depth of Binary Tree (Easy) — Tree depth
- [ ] 543. Diameter of Binary Tree (Easy) — Tree property
- [ ] 226. Invert Binary Tree (Easy) — Tree manipulation
- [ ] 236. Lowest Common Ancestor BT (Medium) — Tree ancestor

Chapter 16: Heaps

- [ ] 703. Kth Largest in Stream (Easy) — Min heap
- [ ] 1046. Last Stone Weight (Easy) — Max heap
- [ ] 692. Top K Frequent Words (Medium) — Custom comparator
- [ ] 295. Find Median from Data Stream (Hard) — Two heaps
- [ ] 502. IPO (Hard) — Greedy + heap
- [ ] 621. Task Scheduler (Medium) — Priority scheduling
- [ ] 355. Design Twitter (Medium) — Heap design

Chapter 17: Tries

- [ ] 208. Implement Trie (Medium) — Trie implementation
- [ ] 211. Design Add and Search Words (Medium) — Trie with wildcard
- [ ] 212. Word Search II (Hard) — Trie + backtracking
- [ ] 14. Longest Common Prefix (Easy) — Trie application
- [ ] 648. Replace Words (Medium) — Trie lookup
- [ ] 677. Map Sum Pairs (Medium) — Trie with values
- [ ] 720. Longest Word in Dictionary (Medium) — Trie building
- [ ] 1268. Search Suggestions System (Medium) — Autocomplete

Chapter 18: Graphs

- [ ] 200. Number of Islands (Medium) — DFS/BFS basics
- [ ] 695. Max Area of Island (Medium) — DFS application
- [ ] 133. Clone Graph (Medium) — Graph traversal
- [ ] 207. Course Schedule (Medium) — Cycle detection
- [ ]

- [ ] 210. Course Schedule II (Medium) — Topological sort
- [ ] 417. Pacific Atlantic Water Flow (Medium) — Multi-source DFS
- [ ] 994. Rotting Oranges (Medium) — Multi-source BFS
- [ ] 542. 01 Matrix (Medium) — BFS shortest path
- [ ] 127. Word Ladder (Hard) — BFS transformation
- [ ] 130. Surrounded Regions (Medium) — DFS boundary
- [ ] 797. All Paths From Source to Target (Medium) — DFS path finding
- [ ] 743. Network Delay Time (Medium) — Dijkstra's algorithm
- [ ] 787. Cheapest Flights Within K Stops (Medium) — Modified Dijkstra
- [ ] 1584. Min Cost to Connect All Points (Medium) — Minimum spanning tree
- [ ] 778. Swim in Rising Water (Hard) — Dijkstra variant
- [ ] 684. Redundant Connection (Medium) — Union find
- [ ] 547. Number of Provinces (Medium) — Connected components
- [ ] 323. Number of Connected Components (Medium) — Union find basics
- [ ] 261. Graph Valid Tree (Medium) — Cycle detection

Chapter 19: Space Complexity

- [ ] 283. Move Zeroes (Easy) — In-place manipulation
- [ ] 876. Middle of Linked List (Easy) — Space optimization
- [ ] 977. Squares of Sorted Array (Easy) — Space-efficient sorting
- [ ] 48. Rotate Image (Medium) — In-place matrix manipulation
- [ ] 287. Find the Duplicate Number (Medium) — O(1) space constraint
- [ ] 41. First Missing Positive (Hard) — O(1) space challenge

Chapter 20: Optimization

- [ ] 53. Maximum Subarray (Medium) — Kadane's algorithm
- [ ] 121. Best Time to Buy and Sell Stock (Easy) — Greedy approach
- [ ] 122. Best Time to Buy and Sell Stock II (Medium) — Greedy optimization
- [ ] 11. Container With Most Water (Medium) — Two pointers

- [ ] 15. 3Sum (Medium) — Two pointers optimization
- [ ] 42. Trapping Rain Water (Hard) — Multiple approaches
- [ ] 238. Product of Array Except Self (Medium) — Prefix/suffix arrays
- [ ] 152. Maximum Product Subarray (Medium) — Dynamic programming
- [ ] 435. Non-overlapping Intervals (Medium) — Greedy intervals
- [ ] 452. Minimum Number of Arrows (Medium) — Greedy optimization