**DSA Pattern Tracker**

Here's your comprehensive tracker:

**1. Two Pointers**

* **Pointers Converging (from ends)**
  + \* [1. Two Sum II - Input Array Is Sorted](https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/) (Easy)
  + \* [125. Valid Palindrome](https://leetcode.com/problems/valid-palindrome/) (Easy)
  + [167. Two Sum IV - Input is a BST](https://leetcode.com/problems/two-sum-iv-input-is-a-bst/) (Easy - can be solved with two pointers on sorted array from BST traversal)
* **Pointers Moving in Same Direction**
  + \* [26. Remove Duplicates from Sorted Array](https://www.google.com/search?q=https://leetcode.com/problems/remove-duplicates-from-from-sorted-array/) (Easy)
  + \* [283. Move Zeroes](https://leetcode.com/problems/move-zeroes/) (Easy)
  + [80. Remove Duplicates from Sorted Array II](https://leetcode.com/problems/remove-duplicates-from-sorted-array-ii/) (Medium)
* **Fast and Slow Pointers (See separate pattern #3 for more details)**
  + [202. Happy Number](https://leetcode.com/problems/happy-number/) (Easy)
  + [141. Linked List Cycle](https://leetcode.com/problems/linked-list-cycle/) (Easy)

**2. Sliding Window**

* **Fixed Size Window**
  + \* [121. Best Time to Buy and Sell Stock](https://leetcode.com/problems/best-time-to-buy-and-sell-stock/) (Easy)
  + [643. Maximum Average Subarray I](https://leetcode.com/problems/maximum-average-subarray-i/) (Easy)
  + [239. Sliding Window Maximum](https://leetcode.com/problems/sliding-window-maximum/) (Hard - requires a deque, but core concept is fixed window)
* **Variable Size Window**
  + \* [3. Longest Substring Without Repeating Characters](https://leetcode.com/problems/longest-substring-without-repeating-characters/) (Medium)
  + \* [209. Minimum Size Subarray Sum](https://leetcode.com/problems/minimum-size-subarray-sum/) (Medium)
  + [713. Subarray Product Less Than K](https://leetcode.com/problems/subarray-product-less-than-k/) (Medium)
* **Window with Auxiliary Data Structures**
  + \* [567. Permutation in String](https://leetcode.com/problems/permutation-in-string/) (Medium)
  + [424. Longest Repeating Character Replacement](https://leetcode.com/problems/longest-repeating-character-replacement/) (Medium)
  + [76. Minimum Window Substring](https://leetcode.com/problems/minimum-window-substring/) (Hard - classic, requires character frequency map)

**3. Fast and Slow Pointers (Hare & Tortoise)**

* **Cycle Detection**
  + \* [141. Linked List Cycle](https://leetcode.com/problems/linked-list-cycle/) (Easy)
  + [202. Happy Number](https://leetcode.com/problems/happy-number/) (Easy - uses cycle detection logic)
* **Finding Cycle Start/Length**
  + \* [142. Linked List Cycle II](https://leetcode.com/problems/linked-list-cycle-ii/) (Medium)
  + [287. Find the Duplicate Number](https://leetcode.com/problems/find-the-duplicate-number/) (Medium - cycle detection in an array)
* **Middle of Linked List**
  + \* [876. Middle of the Linked List](https://leetcode.com/problems/middle-of-the-linked-list/) (Easy)
  + [19. Remove Nth Node From End of List](https://leetcode.com/problems/remove-nth-node-from-end-of-list/) (Medium - can be solved with fast/slow)

**4. Merge Intervals**

* **Simple Merging**
  + \* [56. Merge Intervals](https://leetcode.com/problems/merge-intervals/) (Medium)
  + [228. Summary Ranges](https://leetcode.com/problems/summary-ranges/) (Easy - similar logic of merging consecutive numbers)
* **Insert Interval**
  + \* [57. Insert Interval](https://leetcode.com/problems/insert-interval/) (Medium)
* **Intersection/Disjoint Intervals**
  + \* [986. Interval List Intersections](https://leetcode.com/problems/interval-list-intersections/) (Medium)
  + [252. Meeting Rooms](https://leetcode.com/problems/meeting-rooms/) (Easy - LeetCode Premium, check for overlaps)
  + [253. Meeting Rooms II](https://leetcode.com/problems/meeting-rooms-ii/) (Medium - LeetCode Premium, uses a min-heap)

**5. Cyclic Sort**

* **Finding Missing Number**
  + \* [268. Missing Number](https://leetcode.com/problems/missing-number/) (Easy)
  + [1539. Kth Missing Positive Number](https://leetcode.com/problems/kth-missing-positive-number/) (Easy - can be optimized with binary search, but cyclic sort logic is applicable for finding misses)
* **Finding Duplicate Numbers**
  + \* [287. Find the Duplicate Number](https://leetcode.com/problems/find-the-duplicate-number/) (Medium - often solved with cycle detection, but cyclic sort is another approach)
* **Finding All Missing/Duplicate Numbers**
  + \* [448. Find All Numbers Disappeared in an Array](https://leetcode.com/problems/find-all-numbers-disappeared-in-an-array/) (Easy)
  + \* [442. Find All Duplicates in an Array](https://leetcode.com/problems/find-all-duplicates-in-an-array/) (Medium)

**6. In-place Reversal of a Linked List**

* **Reverse a Singly Linked List**
  + \* [206. Reverse Linked List](https://leetcode.com/problems/reverse-linked-list/) (Easy)
* **Reverse a Sub-list**
  + \* [92. Reverse Linked List II](https://leetcode.com/problems/reverse-linked-list-ii/) (Medium)
* **Reverse Every K-nodes**
  + \* [25. Reverse Nodes in k-Group](https://leetcode.com/problems/reverse-nodes-in-k-group/) (Hard)
* **Reverse Alternate K-nodes**
  + [24. Swap Nodes in Pairs](https://leetcode.com/problems/swap-nodes-in-pairs/) (Medium - specific case of reversing every 2 nodes)

**7. Tree BFS (Level Order Traversal)**

* **Basic Level Order**
  + \* [102. Binary Tree Level Order Traversal](https://leetcode.com/problems/binary-tree-level-order-traversal/) (Medium)
  + [107. Binary Tree Level Order Traversal II](https://leetcode.com/problems/binary-tree-level-order-traversal-ii/) (Easy - just reverse the result)
* **ZigZag Traversal**
  + \* [103. Binary Tree Zigzag Level Order Traversal](https://leetcode.com/problems/binary-tree-zigzag-level-order-traversal/) (Medium)
* **Level Averages/Sums**
  + \* [637. Average of Levels in Binary Tree](https://leetcode.com/problems/average-of-levels-in-binary-tree/) (Easy)
  + [1161. Maximum Level Sum of a Binary Tree](https://leetcode.com/problems/maximum-level-sum-of-a-binary-tree/) (Medium)
* **Connect Level Siblings**
  + \* [116. Populating Next Right Pointers in Each Node](https://leetcode.com/problems/populating-next-right-pointers-in-each-node/) (Medium)
  + [117. Populating Next Right Pointers in Each Node II](https://leetcode.com/problems/populating-next-right-pointers-in-each-node-ii/) (Medium - handles non-perfect binary trees)

**8. Tree DFS (Preorder, Inorder, Postorder Traversal)**

* **Standard Traversals**
  + \* [94. Binary Tree Inorder Traversal](https://leetcode.com/problems/binary-tree-inorder-traversal/) (Easy)
  + \* [144. Binary Tree Preorder Traversal](https://leetcode.com/problems/binary-tree-preorder-traversal/) (Easy)
  + \* [145. Binary Tree Postorder Traversal](https://leetcode.com/problems/binary-tree-postorder-traversal/) (Easy)
* **Path Sum/Sum of Paths**
  + \* [112. Path Sum](https://leetcode.com/problems/path-sum/) (Easy)
  + \* [113. Path Sum II](https://leetcode.com/problems/path-sum-ii/) (Medium)
  + [437. Path Sum III](https://leetcode.com/problems/path-sum-iii/) (Medium - involves hash map for prefix sums)
  + LC 129
  + LC 124
* **Tree Diameter/Max Depth**
  + \* [543. Diameter of Binary Tree](https://leetcode.com/problems/diameter-of-binary-tree/) (Easy)
  + \* [104. Maximum Depth of Binary Tree](https://leetcode.com/problems/maximum-depth-of-binary-tree/) (Easy)
* **Tree Construction from Traversals**
  + \* [105. Construct Binary Tree from Preorder and Inorder Traversal](https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal/) (Medium)

**9. Two Heaps**

* **Finding Median of a Stream**
  + \* [295. Find Median from Data Stream](https://leetcode.com/problems/find-median-from-data-stream/) (Hard - classic Two Heaps problem)
* **Sliding Window Median**
  + [480. Sliding Window Median](https://leetcode.com/problems/sliding-window-median/) (Hard - combines sliding window with two heaps)

**10. Subsets**

* **Generating All Subsets**
  + \* [78. Subsets](https://leetcode.com/problems/subsets/) (Medium)
* **Subsets with Duplicates**
  + \* [90. Subsets II](https://leetcode.com/problems/subsets-ii/) (Medium)
* **Letter Case Permutations**
  + \* [784. Letter Case Permutation](https://leetcode.com/problems/letter-case-permutation/) (Medium - can be seen as a subset-like problem on characters)
* **Combinations/Combinations Sum**
  + \* [39. Combination Sum](https://leetcode.com/problems/combination-sum/) (Medium - unbounded choices)
  + \* [40. Combination Sum II](https://leetcode.com/problems/combination-sum-ii/) (Medium - distinct combinations, each number used once)
  + \* [77. Combinations](https://leetcode.com/problems/combinations/) (Medium)
  + [216. Combination Sum III](https://leetcode.com/problems/combination-sum-iii/) (Medium)

**11. Modified Binary Search**

* **Search in Rotated Sorted Array**
  + \* [33. Search in Rotated Sorted Array](https://leetcode.com/problems/search-in-rotated-sorted-array/) (Medium)
  + [153. Find Minimum in Rotated Sorted Array](https://leetcode.com/problems/find-minimum-in-rotated-sorted-array/) (Medium)
* **Finding First/Last Occurrence**
  + \* [34. Find First and Last Position of Element in Sorted Array](https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/) (Medium)
* **Finding Ceiling/Floor of a Number**
  + [35. Search Insert Position](https://leetcode.com/problems/search-insert-position/) (Easy - finding insertion point, which is ceiling if target not present)
  + (No direct LeetCode problem for strict floor/ceiling by name, but the logic is applied in other problems like finding next greater/smaller element in sorted array)
* **Bitonic Array Search**
  + [852. Peak Index in a Mountain Array](https://leetcode.com/problems/peak-index-in-a-mountain-array/) (Easy)
* **Minimum Difference Element**
  + [Find Element in a Sorted Array with Minimum Difference](https://www.google.com/search?q=https://www.geeksforgeeks.org/find-element-sorted-array-minimum-difference-given-key/) (GFG, good practice problem)

**12. Bitwise XOR**

* **Finding Single Number**
  + \* [136. Single Number](https://leetcode.com/problems/single-number/) (Easy)
  + [137. Single Number II](https://leetcode.com/problems/single-number-ii/) (Medium - numbers appear three times)
* **Finding Two Unique Numbers**
  + \* [260. Single Number III](https://leetcode.com/problems/single-number-iii/) (Medium)
* **Missing Number in Range**
  + \* [268. Missing Number](https://leetcode.com/problems/missing-number/) (Easy - also covered in Cyclic Sort, but XOR is another approach)
* **Flipping Bits/Counting Bits**
  + [191. Number of 1 Bits](https://leetcode.com/problems/number-of-1-bits/) (Easy)
  + [338. Counting Bits](https://leetcode.com/problems/counting-bits/) (Easy)

**13. Top K Elements**

* **K Largest/Smallest Numbers**
  + \* [215. Kth Largest Element in an Array](https://leetcode.com/problems/kth-largest-element-in-an-array/) (Medium - can use Quickselect or Min-Heap)
  + [703. Kth Largest Element in a Stream](https://leetcode.com/problems/kth-largest-element-in-a-stream/) (Easy - direct application of min-heap)
* **K Most/Least Frequent Elements**
  + \* [347. Top K Frequent Elements](https://leetcode.com/problems/top-k-frequent-elements/) (Medium)
  + [451. Sort Characters By Frequency](https://leetcode.com/problems/sort-characters-by-frequency/) (Medium - similar logic, just sort all by frequency)
* **Closest Points to Origin**
  + \* [973. K Closest Points to Origin](https://leetcode.com/problems/k-closest-points-to-origin/) (Medium)

**14. K-way Merge**

* **Merge K Sorted Lists**
  + \* [23. Merge k Sorted Lists](https://leetcode.com/problems/merge-k-sorted-lists/) (Hard - classic problem)
* **Kth Smallest Number in M Sorted Lists**
  + [373. Find K Pairs with Smallest Sums](https://leetcode.com/problems/find-k-pairs-with-smallest-sums/) (Medium - can be adapted to k-way merge logic)
  + (Implicit in many problems where you need to pick the smallest from multiple sources, e.g., finding k smallest elements from sum of two arrays)
* **Kth Smallest Element in a Sorted Matrix**
  + \* [378. Kth Smallest Element in a Sorted Matrix](https://leetcode.com/problems/kth-smallest-element-in-a-sorted-matrix/) (Medium - can be solved with a min-heap, treating rows as sorted lists)

**15. Topological Sort (Graph)**

* **Course Schedule/Dependency Resolution**
  + \* [207. Course Schedule](https://leetcode.com/problems/course-schedule/) (Medium)
  + \* [210. Course Schedule II](https://leetcode.com/problems/course-schedule-ii/) (Medium - return the order)
* **Alien Dictionary**
  + \* [269. Alien Dictionary](https://leetcode.com/problems/alien-dictionary/) (Hard - LeetCode Premium, classic problem)

**16. Dynamic Programming [** [**https://youtu.be/oBt53YbR9Kk?si=OdBPYa6xQqgQVoOx**](https://youtu.be/oBt53YbR9Kk?si=OdBPYa6xQqgQVoOx) **]**

* **0/1 Knapsack**
  + [416. Partition Equal Subset Sum](https://leetcode.com/problems/partition-equal-subset-sum/) (Medium - variation of subset sum)
  + [494. Target Sum](https://leetcode.com/problems/target-sum/) (Medium - can be solved with DP, counts subsets with sum)
* **Unbounded Knapsack (Coin Change)**
  + \* [322. Coin Change](https://leetcode.com/problems/coin-change/) (Medium)
  + \* [518. Coin Change II](https://leetcode.com/problems/coin-change-ii/) (Medium - number of ways)
* **Longest Common Subsequence/Substring**
  + \* [1143. Longest Common Subsequence](https://leetcode.com/problems/longest-common-subsequence/) (Medium)
  + [712. Minimum ASCII Delete Sum for Two Strings](https://leetcode.com/problems/minimum-ascii-delete-sum-for-two-strings/) (Medium - variation of LCS)
  + LC 583
* **Fibonacci/Memoization**
  + \* [509. Fibonacci Number](https://leetcode.com/problems/fibonacci-number/) (Easy)
  + \* [70. Climbing Stairs](https://leetcode.com/problems/climbing-stairs/) (Easy)
* **Kadane's Algorithm**
  + \* [53. Maximum Subarray](https://leetcode.com/problems/maximum-subarray/) (Medium)
* **Grid DP**
  + \* [62. Unique Paths](https://leetcode.com/problems/unique-paths/) (Medium)
  + \* [64. Minimum Path Sum](https://leetcode.com/problems/minimum-path-sum/) (Medium)

**17. Greedy Algorithms [** [**https://youtu.be/bC7o8P\_Ste4?si=13eDgKeXzuCw1CVg**](https://youtu.be/bC7o8P_Ste4?si=13eDgKeXzuCw1CVg%20) **]**

* **Activity Selection**
  + [452. Minimum Number of Arrows to Burst Balloons](https://leetcode.com/problems/minimum-number-of-arrows-to-burst-balloons/) (Medium - similar to activity selection/interval scheduling)
  + LC 455
* **Minimum Spanning Tree (Prim's/Kruskal's)**
  + [1584. Min Cost to Connect All Points](https://leetcode.com/problems/min-cost-to-connect-all-points/) (Medium - can use Kruskal's or Prim's)
* **Dijkstra's Algorithm (Shortest Path)**
  + \* [743. Network Delay Time](https://leetcode.com/problems/network-delay-time/) (Medium)
* **Other Greedy Choices**
  + \* [55. Jump Game](https://leetcode.com/problems/jump-game/) (Medium)
  + [45. Jump Game II](https://leetcode.com/problems/jump-game-ii/) (Medium)

**18. Backtracking**

* **Permutations**
  + \* [46. Permutations](https://leetcode.com/problems/permutations/) (Medium)
  + \* [47. Permutations II](https://leetcode.com/problems/permutations-ii/) (Medium - with duplicates)
* **N-Queens Problem**
  + \* [51. N-Queens](https://leetcode.com/problems/n-queens/) (Hard)
* **Sudoku Solver**
  + \* [37. Sudoku Solver](https://leetcode.com/problems/sudoku-solver/) (Hard)
* **Parentheses Generation**
  + \* [22. Generate Parentheses](https://leetcode.com/problems/generate-parentheses/) (Medium)

**19. Graphs (BFS/DFS Applications)**

* **Connected Components / Island Problems**
  + \* [200. Number of Islands](https://leetcode.com/problems/number-of-islands/) (Medium - classic BFS/DFS grid problem)
  + [695. Max Area of Island](https://leetcode.com/problems/max-area-of-island/) (Medium)
* **Shortest Path in Unweighted Graph (BFS)**
  + \* [1091. Shortest Path in Binary Matrix](https://leetcode.com/problems/shortest-path-in-binary-matrix/) (Medium)
  + [864. Shortest Path to Get All Keys](https://leetcode.com/problems/shortest-path-to-get-all-keys/) (Hard - BFS with bitmasking)
* **Cycle Detection**
  + \* [207. Course Schedule](https://leetcode.com/problems/course-schedule/) (Medium - detecting cycle in directed graph for topo sort)
  + [785. Is Graph Bipartite?](https://leetcode.com/problems/is-graph-bipartite/) (Medium - uses BFS/DFS for coloring, can detect odd cycles)
* **Flood Fill**
  + \* [733. Flood Fill](https://leetcode.com/problems/flood-fill/) (Easy)

**20. Disjoint Set Union (DSU) / Union-Find**

* **Connecting Components**
  + \* [323. Number of Connected Components in an Undirected Graph](https://leetcode.com/problems/number-of-connected-components-in-an-undirected-graph/) (Medium - LeetCode Premium, classic)
  + [130. Surrounded Regions](https://leetcode.com/problems/surrounded-regions/) (Medium - can be solved with DSU or BFS/DFS)
* **Cycle Detection in Undirected Graphs**
  + [Graph with cycles problem](https://www.google.com/search?q=https://www.geeksforgeeks.org/detect-cycle-undirected-graph-using-union-find/) (GFG - conceptual problem, often embedded in other problems)
* **Graph Connectivity Problems**
  + \* [990. Satisfiability of Equality Equations](https://leetcode.com/problems/satisfiability-of-equality-equations/) (Medium)
  + [261. Graph Valid Tree](https://leetcode.com/problems/graph-valid-tree/) (Medium - LeetCode Premium, combines DSU with graph properties)