

# LeetCode Problem Categories by Technique

## 1. Array Manipulation

**Techniques:** Sliding window, two pointers, prefix sums, hashing.

- Two Sum
- Maximum Subarray
- Product of Array Except Self
- Subarray Sum Equals K
- Merge Intervals

## 2. Strings

**Techniques:** String manipulation, pattern matching, two pointers, hashing.

- Longest Substring Without Repeating Characters
- Valid Anagram
- Group Anagrams
- Longest Palindromic Substring
- String to Integer (atoi)

## 3. Linked List

**Techniques:** Slow and fast pointers, dummy nodes, reversing a list.

- Reverse Linked List
- Merge Two Sorted Lists
- Linked List Cycle
- Add Two Numbers
- Intersection of Two Linked Lists

## 4. Stack and Queue

**Techniques:** Monotonic stack, breadth-first search (BFS), depth-first search (DFS).

- Valid Parentheses
- Daily Temperatures
- Evaluate Reverse Polish Notation
- Sliding Window Maximum
- Implement Stack Using Queues

## 5. Binary Tree

**Techniques:** DFS, BFS, recursion, iterative traversal.

- Binary Tree Inorder Traversal

- Maximum Depth of Binary Tree
- Path Sum
- Lowest Common Ancestor of a Binary Tree
- Serialize and Deserialize Binary Tree

## 6. Binary Search Tree

**Techniques:** BST properties, recursion, in-order traversal.

- Validate Binary Search Tree
- Insert into a Binary Search Tree
- Kth Smallest Element in a BST
- Lowest Common Ancestor of a BST
- Binary Search Tree Iterator

## 7. Recursion and Backtracking

**Techniques:** Generate permutations/combinations, search space pruning.

- Subsets
- Permutations
- Combination Sum
- Sudoku Solver
- Word Search

## 8. Dynamic Programming

**Techniques:** State definition, transitions, memoization, tabulation.

- Climbing Stairs
- House Robber
- Longest Increasing Subsequence
- Coin Change
- Longest Common Subsequence

## 9. Greedy

**Techniques:** Local optimization leads to global solution.

- Jump Game
- Gas Station
- Meeting Rooms II
- Partition Labels
- Minimum Number of Arrows to Burst Balloons

## 10. Binary Search

**Techniques:** Divide and conquer, searching in sorted arrays.

- Binary Search
- Search in Rotated Sorted Array
- Find Minimum in Rotated Sorted Array
- Koko Eating Bananas
- Median of Two Sorted Arrays

## 11. Graphs

**Techniques:** BFS, DFS, union-find, shortest path algorithms.

- Number of Islands
- Clone Graph
- Course Schedule
- Minimum Spanning Tree
- Dijkstra's Algorithm

## 12. Bit Manipulation

**Techniques:** Bitwise operations, XOR, masking.

- Single Number
- Hamming Distance
- Counting Bits
- Reverse Bits
- Subsets (Bitmasking)

## 13. Sliding Window

**Techniques:** Dynamic window resizing, maintaining invariants.

- Minimum Window Substring
- Longest Substring with At Most Two Distinct Characters
- Sliding Window Maximum
- Subarray Product Less Than K
- Find All Anagrams in a String

## 14. Two Pointers

**Techniques:** Moving two pointers towards each other or independently.

- Two Sum II (Input Array Is Sorted)
- Container With Most Water
- Three Sum
- Remove Duplicates from Sorted Array
- Trapping Rain Water

## 15. Sorting

**Techniques:** Merge sort, quicksort, custom comparators.

- Merge Intervals
- Sort Colors
- Kth Largest Element in an Array
- Meeting Rooms
- Largest Number

## 16. Heap (Priority Queue)

**Techniques:** Min-heap, max-heap, custom comparators.

- Kth Largest Element in a Stream
- Top K Frequent Elements
- Find Median from Data Stream
- Merge K Sorted Lists
- Meeting Rooms II

## 17. Math and Number Theory

**Techniques:** Modular arithmetic, combinatorics, prime numbers.

- Fibonacci Number
- Greatest Common Divisor of Strings
- Sqrt(x)
- Pow(x, n)
- Count Primes

## 18. Union-Find (Disjoint Set)

**Techniques:** Path compression, union by rank.

- Number of Connected Components in an Undirected Graph
- Redundant Connection
- Graph Valid Tree
- Accounts Merge
- Smallest String With Swaps

## 19. Trie (Prefix Tree)

**Techniques:** Word prefix storage and retrieval.

- Implement Trie (Prefix Tree)
- Add and Search Word
- Word Search II
- Replace Words
- Maximum XOR of Two Numbers in an Array

## 20. Interval Problems

**Techniques:** Sorting intervals, merging intervals.

- Merge Intervals
- Insert Interval
- Non-Overlapping Intervals
- Meeting Rooms
- Minimum Number of Arrows to Burst Balloons

## 21. Divide and Conquer

**Techniques:** Recursive problem decomposition.

- Merge Sort
- Quick Sort
- Search in a Rotated Sorted Array
- Median of Two Sorted Arrays
- Maximum Subarray (Divide and Conquer Version)

## Study Plan Calendar

**Study Plan:** Starting from **14/12/2024** to **21/4/2025**. The study plan is organized into continuous intervals, with one-week revisions after completing a topic.

1. **Week 1 (14/12/2024 - 20/12/2024):** Array Manipulation
2. **Week 2 (21/12/2024 - 27/12/2024):** Strings
3. **Week 3 (28/12/2024 - 03/01/2025):** Linked List
4. **Week 4 (04/01/2025 - 10/01/2025):** Stack and Queue
5. **Week 5 (11/01/2025 - 17/01/2025):** Binary Tree
6. **Week 6 (18/01/2025 - 24/01/2025):** Binary Search Tree
7. **Week 7 (25/01/2025 - 31/01/2025):** Recursion and Backtracking
8. **Week 8 (01/02/2025 - 07/02/2025):** Dynamic Programming
9. **Week 9 (08/02/2025 - 14/02/2025):** Greedy
10. **Week 10 (15/02/2025 - 21/02/2025):** Binary Search
11. **Week 11 (22/02/2025 - 28/02/2025):** Graphs
12. **Week 12 (01/03/2025 - 07/03/2025):** Bit Manipulation
13. **Week 13 (08/03/2025 - 14/03/2025):** Sliding Window
14. **Week 14 (15/03/2025 - 21/03/2025):** Two Pointers
15. **Week 15 (22/03/2025 - 28/03/2025):** Sorting
16. **Week 16 (29/03/2025 - 04/04/2025):** Heap (Priority Queue)
17. **Week 17 (05/04/2025 - 11/04/2025):** Math and Number Theory
18. **Week 18 (12/04/2025 - 18/04/2025):** Union-Find (Disjoint Set)
19. **Week 19 (19/04/2025 - 21/04/2025):** Trie (Prefix Tree)
20. Revisions will take place weekly after completing each topic.