1. Fast and Slow Pointer

Description: This technique uses two pointers moving at different speeds to solve problems involving cycles, such as finding the middle of a list, detecting loops, or checking for palindromes.

- Linked List Cycle II
- Remove nth Node from the End of List
- Find the Duplicate Number
- Palindrome Linked List

2. Overlapping Intervals

Description: Intervals are often manipulated through sorting and merging based on their start and end times.

- Basic Merge: Merge Intervals
- Interval Insertion: Insert Interval
- Minimum Number of Arrows to Burst Balloons
- Meeting Rooms ii
- Non-overlapping Intervals

3. Sliding Window

Description: A sliding window is a subarray or substring that moves over data to solve problems efficiently in linear time.

Fixed Size

- Maximum Sum Subarray of Size K
- Number of Subarrays having Average Greater or Equal to Threshold
- Sliding Subarray Beauty
- Permutation in String
- Sliding Window Maximum

Variable Size

- Longest Substring Without Repeating Characters
- Minimum Size Subarray Sum
- Subarray Product Less Than K
- Max Consecutive Ones
- Fruits Into Baskets
- Count Number of Nice Subarrays
- Minimum Window Substring: Minimum Window Substring

4. Two Pointers

Description: The two pointers technique involves having two different indices move through the input at different speeds to solve various array or linked list problems.

- Two Sum II - Input Array is Sorted
- Dutch National Flag: Sort Colors
- Next Permutation
- Bag of Tokens
- Container with most water
- Trapping Rain Water

5. Cyclic Sort (Index-Based)

Description: Cyclic sort is an efficient approach to solve problems where numbers are consecutively ordered and must be placed in the correct index.

- Missing Number
- Find Missing Numbers
- Set Mismatch
- First Missing Positive

6. Reversal of Linked List (In-place)

Description: Reversing a linked list in place without using extra space is key for problems that require in-place list manipulations.

- Reverse Linked List
- Reverse Nodes in k-Group
- Swap Nodes in Pairs

7. Matrix Manipulation

Description: Problems involving 2D arrays (matrices) are often solved using row-column traversal or manipulation based on matrix properties.

- Rotate Image
- Spiral Matrix
- Set Matrix Zeroes
- Game of Life

8. Breadth First Search (BFS)

Description: BFS explores nodes level by level using a queue. It is particularly useful for shortest path problems.

- Shortest Path in Binary Matrix
- Rotten Oranges
- As Far From Land as Possible
- Word Ladder: Word Ladder

9. Depth First Search (DFS)

Description: DFS explores as far as possible along a branch before backtracking. It's useful for graph traversal, pathfinding, and connected components.

- Number of Closed Islands
- Coloring a Border
- DFS from boundary: Number of Enclaves
- Shortest time: Time Needed to Inform all Employees
- Cyclic Find: Find Eventual Safe States

10. Backtracking

Description: Backtracking helps in problems where you need to explore all potential solutions, such as solving puzzles, generating combinations, or finding paths.

- Permutation ii
- Combination Sum
- Generate Parenthesis
- N-Queens
- Sudoku Solver
- Palindrome Partitioning
- Word Search: Word Search

11. Modified Binary Search

Description: A modified version of binary search that applies to rotated arrays, unsorted arrays, or specialized conditions.

- Search in Rotated Sorted Array
- Find Minimum in Rotated Sorted Array
- Find Peak Element
- Minimum Time to Arrive on Time
- Capacity to Ship Packages within 'd' Days

- Koko Eating Bananas
- Find in Mountain Array
- Median of Two Sorted Arrays

12. Bitwise XOR

Description: XOR is a powerful bitwise operator that can solve problems like finding single numbers or efficiently pairing elements.

- Missing Number
- Single Number ||
- Single Number III
- Find the Original array of Prefix XOR
- XOR Queries of a Subarray

13. Top 'K' Elements

Description: This pattern uses heaps or quickselect to efficiently find the top 'K' largest/smallest elements from a dataset.

- Top K Frequent Elements
- Kth Largest Element
- Ugly Number ii
- K Closest Points to Origin

14. K-way Merge

Description: The K-way merge technique uses a heap to efficiently merge multiple sorted lists or arrays.

- Find K Pairs with Smallest Sums
- Kth Smallest Element in a Sorted Matrix
- Merge K Sorted Lists
- Smallest Range: Smallest Range Covering Elements from K Lists

15. Two Heaps

Description: This pattern uses two heaps (max heap and min heap) to solve problems involving tracking medians and efficiently managing dynamic data.

- Find Median from Data Stream
- Sliding Window Median

- IPO

16. Monotonic Stack

Description: A monotonic stack helps solve range queries by maintaining a stack of elements in increasing or decreasing order.

- Next Greater Element II
- Next Greater Node in Linked List
- Daily Temperatures
- Online Stock Span
- Maximum Width Ramp
- Largest Rectangle in Histogram

17. **Trees**

Level Order Traversal (BFS in Binary Tree)

- Level order Traversal
- Zigzag Level order Traversal
- Even Odd Tree
- Reverse odd Levels
- Deepest Leaves Sum
- Add one row to Tree
- Maximum width of Binary Tree
- All Nodes Distance K in Binary tree

Tree Construction

- Construct BT from Preorder and Inorder
- Construct BT from Postorder and Inorder
- Maximum Binary Tree
- Construct BST from Preorder

Height related Problems

- Maximum Depth of BT
- Balanced Binary Tree
- Diameter of Binary Tree
- Minimum Depth of BT
- Binary Tree Maximum Path Sum

Root to leaf path problems

- Binary Tree Paths
- Path Sum ii
- Sum Root to Leaf numbers
- Smallest string starting from Leaf
- Insufficient nodes in root to Leaf
- Pseudo-Palindromic Paths in a Binary Tree

Ancestor problem

- LCA of Binary Tree
- Maximum difference between node and ancestor
- LCA of deepest leaves
- Kth Ancestor of a Tree Node

Binary Search Tree

- Validate BST
- Range Sum of BST
- Minimum Absolute Difference in BST
- Insert into a BST
- LCA of BST

18. DYNAMIC PROGRAMMING

Take / Not take (DP)

Description: Solve optimization problems like selecting items with the max/min value under certain constraints.

- House Robber ii
- Target Sum
- Partition Equal Subset Sum
- Ones and Zeroes
- Last Stone Weight ii

Infinite Supply (DP)

Description: Similar to the 0/1 knapsack, but items can be chosen multiple times.

- Coin Change
- Coin Change II
- Perfect Squares
- Minimum Cost For Tickets

Longest Increasing subsequence

Description: It involves finding the longest subsequence of a given sequence where the elements are in ascending order

- Longest Increasing Subsequence
- Largest Divisible Subset
- Maximum Length of Pair Chain
- Number of LIS
- Longest String Chain

DP on Grids

Description: Dynamic Programming on matrices involves solving problems that can be broken down into smaller overlapping subproblems within a matrix.

- Unique Paths ii
- Minimum Path Sum
- Triangle
- Minimum Falling Path Sum
- Maximal Square
- Cherry Pickup
- Dungeon Game: Dungeon Game

DP on Strings

Description: It Involves 2 strings, whenever you are considering two substrings/subsequence from given two strings, concentrate on what happens when the last characters of the two substrings are same, i.e, matching.

- Longest Common Subsequence
- Longest Palindromic Subsequence
- Palindromic Substrings
- Longest Palindromic Substrings
- Edit Distance
- Minimum ASCII Delete Sum for Two Strings
- Distinct Subsequences
- Shortest Common Supersequence
- Wildcard Matching

DP on Stocks

Description: It focuses on maximizing profit from buying and selling stocks over time while considering constraints.

- Buy and Sell Stocks ii
- Buy and Sell Stocks iii
- Buy and Sell Stocks iv
- Buy and Sell Stocks with Cooldown
- Buy and Sell Stocks with Transaction fee

Partition DP (MCM)

Description: It Involves a sequence that needs to be divided into partitions in an optimal way. The goal is often to minimize or maximize a cost function, such as computation time, multiplications, or some other metric, by exploring all possible partitions and combining results from subproblems.

- Partition array for Maximum Sum

- Burst Balloons
- Minimum Cost to Cut a Stick
- Palindrome Partitioning ii

19. **Graphs**

Topological Sort

Description: Topological sorting is useful for tasks that require dependency resolution (InDegree) in directed acyclic graphs (DAGs).

- Course Schedule
- Course Schedule II
- Sequence Reconstruction
- Alien Dictionary

Union Find (Disjoint Set)

Description: Union-Find (or Disjoint Set) is used to solve problems involving connectivity or grouping, often in graphs.

- Number of Operations to Make Network Connected
- Redundant Connection
- Accounts Merge
- Satisfiability of Equality Equations

Graph Algorithms

Description: Advanced graph algorithms are used to solve complex problems involving shortest paths, minimum spanning trees, and graph cycles.

- Kruskal's Algorithm: Minimum Cost to connect all Points
- Dijkstra's Algorithm: Cheapest Flights Within K Stops
- Floyd-Warshall: Find the City with Smallest Number of Neighbours at a Threshold Distance
- Bellman Ford: Network Delay time

20. Greedy

Description: Greedy algorithms make local optimal choices at each step, which lead to a global optimal solution for problems like scheduling and resource allocation.

- Jump Game ii
- Gas Station
- Bag of Tokens
- Boats to Save People
- Wiggle Subsequence

- Car Pooling
- Candy

21. Design Data Structure

Description: It involves building custom data structures to efficiently handle specific operations, like managing data access, updates, and memory usage. Focusing on optimizing performance and resource management.

- Design Twitter
- Design Browser History
- Design Circular Deque
- Snapshot Array
- LRU Cache
- LFU Cache

Some Useful Articles on LeetCode for Better Understanding!

Two Pointers

- Solved all Two Pointers problems in 100 days

Sliding Window

- Sliding Window Technique and Question Bank
- C++ Maximum Sliding Window Cheatsheet Template!

Greedy

- Greedy for Beginners: Problems & Sample Solutions
- Top Greedy Questions

Linked List

- Become Master In Linked List
- Must-Do LinkedList Problems on LeetCode

Trees

- Tree Question Pattern | 2021 Placement

- Master Tree Patterns

Binary Search

- 5 Variations of Binary Search
- Binary Search for Beginners: Problems & Patterns

Dynamic Programming (DP)

- Dynamic Programming Patterns
- DP for Beginners: Problems & Patterns

Graphs

- Graph For Beginners
- Become Master In Graph
- Graph algorithms + problems to practice

Bit Manipulation

- Bit Manipulation Problem solving
- All Types of Patterns for Bits Manipulations and How to use i