**1. Stack (LIFO - Last In, First Out)**

**Implementation**

* Implement Stack using **Array**
* Implement Stack using **Linked List**
* Implement Stack using **2 Queues**
* Implement Stack using **1 Queue**
* Implement Stack using **Deque**
* Implement **Min Stack** (Stack that supports O(1) min retrieval)
* Implement **Max Stack** (Stack that supports O(1) max retrieval)

**Stack Problems**

✅ **Basic Problems:**

1. **Push, Pop, Peek Operations**
2. **Reverse a Stack using Recursion**
3. **Sort a Stack using Recursion**
4. **Check for Balanced Parentheses** (includes {}, (), [])
5. **Find Next Greater Element**
6. **Find Next Smaller Element**
7. **Find Previous Greater Element**
8. **Find Previous Smaller Element**
9. **Span of Stock Prices**
10. **Largest Rectangle in Histogram**
11. **Evaluate Postfix Expression**
12. **Convert Infix Expression to Postfix**
13. **Convert Infix Expression to Prefix**
14. **Evaluate Prefix Expression**
15. **Check for Redundant Brackets**
16. **Celebrity Problem (Graph + Stack)**
17. **Simplify Directory Path (/a/./b/../../c/ → /c)**
18. **Remove K Digits to Make Smallest Number**
19. **Decode String (3[a]2[bc] → "aaabcbc")**
20. **Basic Calculator I, II, III (Leetcode)**

✅ **Advanced Problems:** 21. **Find Maximum Area of Binary Matrix (Using Stack)** 22. **Asteroid Collision** 23. **Trapping Rainwater Problem** 24. **Sliding Window Maximum (Using Deque)** 25. **Check if Stack Operations are Valid** 26. **Largest Rectangle Under Skyline** 27. **Largest Valid Substring with Matching Parentheses** 28. **Valid Parenthesis String (\* as wildcard)**