

Chapter 10

Practice Problems

Easy → Medium → Hard progression — 20 curated LeetCode problems

Easy — Build Your Foundation

#	Problem	Difficulty	Key Concept
1	Implement Stack using Array	Easy	Array + top pointer
2	Implement Queue using Array	Easy	Circular array
3	Valid Parentheses (LC 20)	Easy	Stack — push/pop matching
4	Queue using Two Stacks (LC 232)	Easy	Amortized O(1) dequeue
5	Stack using Two Queues (LC 225)	Easy	Reverse FIFO to LIFO
6	Next Greater Element I (LC 496)	Easy	Monotonic stack

Medium — Level Up

#	Problem	Difficulty	Key Concept
1	Min Stack (LC 155)	Medium	Auxiliary min stack
2	Evaluate Reverse Polish Notation (LC 150)	Medium	Operand/operator stack
3	Daily Temperatures (LC 739)	Medium	Monotonic stack (decreasing)

4	Design Circular Queue (LC 622)	Medium	Circular array with front/rear
5	Decode String (LC 394)	Medium	Stack for nested brackets
6	Simplify Path (LC 71)	Medium	Stack — directory navigation
7	Remove K Digits (LC 402)	Medium	Monotonic stack
8	Number of Visible People in Queue (LC 1944)	Medium	Decreasing monotonic stack

■ Hard — Interview Killers

#	Problem	Difficulty	Key Concept
1	Trapping Rain Water (LC 42)	Hard	Monotonic stack or two pointer
2	Largest Rectangle in Histogram (LC 84)	Hard	Monotonic stack (area calc)
3	Basic Calculator (LC 224)	Hard	Stack for +/-/parens
4	Sliding Window Maximum (LC 239)	Hard	Monotonic deque
5	Maximum Frequency Stack (LC 895)	Hard	Stack + frequency map
6	Remove Invalid Parentheses (LC 301)	Hard	BFS using queue

■ Problem Solving Template

When you see a Stack/Queue problem, ask: 1. Do I need to track history? → Stack 2. Do I need to reverse something? → Stack 3. Is there a 'next greater/smaller' pattern? → Monotonic Stack 4. Do I need level-order or shortest path? → Queue (BFS) 5. Can I use the LIFO property to cancel/match? → Stack Then write: declare DS → iterate → push/enqueue condition → pop/dequeue condition → return