# Stack – Core Patterns Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Pattern | Core Idea | Mental Idea | Typical Scenarios |
| 1 | Basic Stack (LIFO Processing) | Use a stack to store elements until they’re no longer needed; pop when processed. | The last thing I saw is the first I’ll need to resolve. | Valid Parentheses, Undo operations, Expression parsing |
| 2 | Monotonic Increasing Stack | Maintain elements in increasing order (top is always the smallest or latest valid). | I want to know when something bigger appears — pop smaller elements as I go. | Next Greater Element, Daily Temperatures, Stock Span |
| 3 | Monotonic Decreasing Stack | Maintain elements in decreasing order (top always holds the largest/most recent). | I want to know when something smaller appears — pop larger elements as I go. | Largest Rectangle in Histogram, Trapping Rain Water |
| 4 | Two-Stack Evaluation (Operators & Operands) | Use one stack for values and another for operators to evaluate expressions. | Hold operands until an operator with higher precedence appears. | Evaluate Reverse Polish Notation, Infix to Postfix conversion |
| 5 | Stack with Index (Tracking Positions) | Store indices (not values) to calculate distances or ranges efficiently. | I need to know where something smaller/larger appeared before. | Largest Rectangle in Histogram, Next Smaller Element |
| 6 | Stack for Recursion Simulation | Explicitly simulate recursion using your own stack data structure. | I can’t use recursion, so I’ll push function calls manually. | DFS iterative, Evaluate binary tree expressions |
| 7 | Stack of Stacks (Composite Behavior) | Manage multiple stacks or bounded stacks with overflow logic. | When one stack fills, push to the next; still maintain stack rules. | Set of Stacks (OO Design pattern), Browser history tabs |
| 8 | Stack with Min/Max Tracking | Augment each stack node with running min/max to get O(1) min/max queries. | Each element remembers the min/max seen so far. | Min Stack, Max Stack |
| 9 | Hybrid (Stack + Queue / Stack + Heap) | Use stack along with another data structure for complex state tracking. | Stack gives me last-seen logic; another structure maintains priority or order. | Implement Queue using Stacks, Online Stock Span |