

Chapter 1

■ Introduction to Stacks & Queues

Why they matter — in interviews, in real life, and in your OS

■ Real-World Analogy

Imagine you're washing dishes. You place plates in a pile — the last plate you put down is the first one you wash. That pile of plates IS a Stack. Now imagine a queue at a movie theatre. The first person in line is the first person who gets a ticket. That line IS a Queue. Two simple physical ideas. Two of the most powerful data structures in computer science.

■ What Are They?

A **Stack** is a linear data structure that follows the **LIFO** principle — **Last In, First Out**. Elements are inserted and removed from the same end, called the **top**.

A **Queue** is a linear data structure that follows the **FIFO** principle — **First In, First Out**. Elements are inserted at the **rear** and removed from the **front**.

■ Stack

- LIFO — Last In, First Out
- Push to top, Pop from top
- Used in: recursion, undo, browsers
- Think: pile of plates

■ Queue

- FIFO — First In, First Out
- Enqueue at rear, Dequeue from front
- Used in: BFS, task scheduling, printers
- Think: ticket counter line

■ Why Do They Matter?

In Interviews:

- ~25% of all DSA interview questions involve stacks or queues
- They appear in almost every company: Google, Amazon, Meta, Microsoft
- They're the backbone of harder topics: graphs (BFS/DFS), dynamic programming, monotonic patterns

In Real Life & Systems:

Type	Real-World Use
Stack	Browser back/forward history
Stack	Function call stack (your OS uses this!)
Stack	Undo/Redo in editors (VS Code, Photoshop)
Stack	Expression evaluation (calculators)
Queue	CPU task scheduling (Round Robin)
Queue	Print queue in your OS
Queue	BFS in Google Maps pathfinding
Queue	Message queues (Kafka, RabbitMQ)

■ **Mental Seed:** Every time your code calls a function, the OS pushes a stack frame. Every time the function returns, it pops. You have been using stacks your entire programming life — now you'll understand them deeply.