

Chapter 11

■ Quick Revision

Everything you need 30 minutes before your interview

■ One-Line Definitions

Term	One-line
Stack	LIFO — last in, first out. Think: plates, browser back, call stack.
Queue	FIFO — first in, first out. Think: ticket line, BFS, task scheduler.
Monotonic Stack	Stack that maintains increasing/decreasing order. Used for NGE/NLE problems.
Deque	Double-ended queue. Can push/pop from both ends. Used for sliding window max.
Circular Queue	Fixed-size queue that wraps using modulo. No wasted space.
Min Stack	Stack + auxiliary stack that tracks current minimum at each level.

■ Java Cheat Sheet

Java Quick Reference

JAVA

```
// Stack 

Deque stack = new ArrayDeque<>();  
stack.push(x); // O(1)


```

```
stack.pop(); // O(1) – throws if empty  
stack.peek(); // O(1) – throws if empty  
stack.isEmpty(); // O(1)  
  
// Queue ████████████████████████████████████████████████████████████████  
Queue queue = new LinkedList<>();  
queue.offer(x); // O(1) – enqueue  
queue.poll(); // O(1) – dequeue, null if empty  
queue.peek(); // O(1) – front, null if empty  
  
// PriorityQueue (Min-Heap) ██████████████████████████████████████████  
PriorityQueue pq = new PriorityQueue<>();  
pq.offer(x); // O(log n)  
pq.poll(); // O(log n) – smallest first  
  
// Deque (Double-ended) ████████████████████████████████████████████  
Deque dq = new ArrayDeque<>();  
dq.offerFirst(x); dq.offerLast(x);  
dq.pollFirst(); dq.pollLast();
```

BFS on graph (V vertices, E edges)	$O(V+E)$	$O(V)$	Visit each node/edge once
Queue using Two Stacks — dequeue	$O(1)$ amortized	$O(n)$	Amortized analysis

■ Interview Tips

- ✓ Always check `isEmpty()` before `pop()/peek()` — never assume non-empty
- ✓ In Java, use `ArrayDeque` for stacks and `LinkedList/ArrayDeque` for queues — never `java.util.Stack`
- ✓ Monotonic stack problems: decide INCREASING or DECREASING based on what you want to track
- ✓ Queue + `HashMap` = the backbone of BFS with visited tracking
- ✓ If a problem says 'most recent' → Stack. If 'first come first serve' → Queue
- ✓ Two stacks simulate queue, two queues simulate stack — know both patterns cold
- ✓ Trapping Rain Water: three approaches — brute $O(n^2)$, prefix arrays $O(n)$, monotonic stack $O(n)$
- ✓ In circular queue: $(rear + 1) \% capacity$ for next index — always use modulo