■ Arrays — Ultimate Interview Guide (Striver A2Z + NeetCode 150)

■ EASY LEVEL

Problem	Pattern / Concept	Key Idea
Find Largest Element in Array	Linear Scan	Track max while iterating
Find Second Largest Element	Two Pass / Single Scan	Track max and second max
Check if Array is Sorted	Simple Traversal	Compare adjacent elements
Remove Duplicates from Sorted Array	Two Pointers	Maintain slow pointer for unique values
Move Zeroes to End	Two Pointers	Shift non-zero elements forward
Rotate Array (by k)	Reverse / Modulo	Reverse parts of array
Contains Duplicate	HashSet	Track visited elements
Missing Number (0 to n)	XOR / Sum	XOR all numbers & indices
Intersection of Two Arrays	HashSet / Two Pointers	Store one array’s elements in set
Union of Two Arrays	Set	Combine unique elements
Single Number	XOR	XOR all elements
Maximum Consecutive Ones	Counting	Count and reset on 0
Leaders in Array	Reverse Traversal	Track running max from right
Left Rotate by One	Simple Shift	Move all elements left by one

■ MEDIUM LEVEL

Problem	Pattern / Concept	Key Idea
Two Sum	Hashing	Check if target - num exists
3Sum	Sorting + Two Pointers	Fix one element, find pair
4Sum	Two Pointers + Sorting	Fix two, use two-pointer sum
Best Time to Buy and Sell Stock	Greedy	Track min price and max profit
Maximum Subarray Sum (Kadane’s)	Dynamic Programming	Track current & global max
Subarray Sum Equals K	Prefix Sum + HashMap	Count prefix sum differences
Product of Array Except Self	Prefix & Suffix Product	Multiply left & right arrays
Majority Element (Boyer-Moore)	Voting Algorithm	Maintain candidate & count
Sort Colors (Dutch Flag)	Three Pointers	Partition 0s, 1s, 2s
Merge Intervals	Sorting + Greedy	Sort and merge overlapping
Insert Interval	Interval Merging	Handle overlapping while inserting
Set Matrix Zeroes	Matrix + Marker	Use first row & col as markers
Spiral Matrix Traversal	Simulation / Boundaries	Traverse layers in spiral order
Rotate Image (Matrix Rotation)	Transpose + Reverse	Rotate 90° in-place
Find Duplicate Number	Floyd’s Cycle Detection	Use slow-fast pointers

Missing and Repeating Number	XOR / Math	Use XOR or equations
Longest Consecutive Sequence	HashSet	Count sequence length from start
Container With Most Water	Two Pointers	Move pointer of smaller height
Trapping Rain Water	Prefix-Suffix / Two Pointers	Track max left & right bars
Next Permutation	Greedy / Two Pointers	Find pivot, swap & reverse suffix
Merge Sorted Arrays (In-place)	Two Pointers (End Merge)	Merge from back to front
Maximum Product Subarray	DP (Min/Max Tracking)	Handle negatives carefully
Subarray with Given XOR	Prefix XOR + HashMap	Count XOR pairs = target
Rearrange Array Alternating +ve -ve	Two Pointers	Place positives & negatives alternately
Kadane’s Circular Subarray	Kadane + Inversion	Max(sum - min_subarray)
Search in Rotated Sorted Array	Binary Search	Determine sorted half
Find Minimum in Rotated Sorted Array	Binary Search	Compare mid with end
Find Peak Element	Binary Search	Compare mid with mid+1

■ HARD LEVEL

Problem	**Pattern / Concept**	**Key Idea**
Median of Two Sorted Arrays	Binary Search on Partitions	Balance left-right halves
Maximum Sum Rectangle in 2D Matrix	Kadane’s 2D	Apply Kadane’s on row sums
Kth Largest Element in Array	Heap / Quickselect	Use min-heap of size k
Count Inversions in Array	Merge Sort	Count during merge step
Majority Element II (n/3 times)	Boyer-Moore Extended	Track 2 candidates
Trapping Rain Water II (3D version)	BFS + Min Heap	Boundary-based filling
Maximum Sum Subarray of Size K	Sliding Window	Maintain fixed window sum
Sliding Window Maximum	Deque / Monotonic Queue	Maintain decreasing deque
Longest Subarray with Sum ≤ K	Prefix + Binary Search	Track prefix sums efficiently
Largest Subarray with 0 Sum	HashMap	Track prefix sums
Minimize Maximum Difference of Heights	Greedy	Adjust min/max by K