

PART 2: ARRAYS — PROPER STRUCTURE

❑ PHASE 0: ARRAY BASICS (START HERE – VERY IMPORTANT)

Tools needed:

- ✓ loops
- ✓ indexing
- ✓ max/min
- ✓ counting

Questions (DO FIRST, IN THIS ORDER)

- Largest Element in an Array
- Second Largest Element in an Array (without sorting)
- Check if the array is sorted
- Linear Search
- Find missing number in an array
- Maximum Consecutive Ones
- Leaders in an Array problem

❑ Why first?

These teach:

- traversal
- boundary handling
- condition logic

If you struggle here → stop and fix basics.

❑ PHASE 1: ARRAY MANIPULATION (NO NEW TOOL)

Still **NO two pointers**, just logic.

Questions

- Left Rotate an array by one place
- Left rotate an array by D places
- Move Zeros to end

- Find the Union
- Pascal's Triangle

Focus on:

- shifting
 - temporary variables
 - index mapping
-

PHASE 2: TWO POINTERS (FIRST REAL TECHNIQUE)

STOP if you haven't studied Two Pointers yet.

Core idea:

- use two indices
- reduce nested loops

Questions

- Remove duplicates from Sorted array
- Sort an array of 0's 1's and 2's
- 2Sum Problem
- Rearrange the array in alternating positive and negative items
- Merge two sorted arrays without extra space

If a question says **sorted array** → think **Two Pointers**

PHASE 3: PREFIX SUM / KADANE

These are where beginners panic. That's normal.

Concepts:

- prefix sum
- cumulative effect
- Kadane's algorithm

Questions

- Kadane's Algorithm (maximum subarray sum)
- Print subarray with maximum subarray sum
- Maximum Subarray with 0 Sum
- Maximum Product Subarray

These are **NOT basic array problems**

They are optimization problems.

PHASE 4: HASHING (ARRAY + MAP)

Now arrays + HashMap.

Concepts:

- frequency counting
- seen-before logic
- prefix sum + hashmap

Questions

- Find the number that appears once, and other numbers twice
- Longest Subarray with given sum K (positives)
- Longest Subarray with sum K (positives + negatives)
- Longest Consecutive Sequence in an Array
- Count subarrays with given sum
- Count number of subarrays with given xor K

If brute force is $O(n^2)$, hashing usually makes it $O(n)$.

PHASE 5: MATRIX (2D ARRAYS)

These need **patience**, not new DS.

Questions

- Set Matrix Zeroes
- Rotate Matrix by 90 degrees
- Print the matrix in spiral manner

Think:

- boundaries
 - row/column control
-

PHASE 6: ADVANCED ARRAY PATTERNS (DO LAST)

These combine **multiple tools**.

Questions

- Majority Element ($> n/2$ times)
- Majority Element ($> n/3$ times)
- Next Permutation
- Stock Buy and Sell
- 3 Sum Problem
- 4 Sum Problem
- Merge Overlapping Subintervals
- Find the repeating and missing number
- Count Inversions
- Reverse Pairs

Do these **ONLY after** phases 0–4.