Phase 1- Basic Array Problems

🟩 Topic 1: Traversal & Basic Logic (5)

1. Left Rotate by 1

Rotate the array by one position to the left.

2. Left Rotate by d

Rotate array by d positions (in-place with no extra space).

3. Mirror Index Element

Print element at the middle index (⌊N/2⌋).

4. Running Prefix Sum

Modify array to hold prefix sums: arr[i] = arr[0] + ... + arr[i].

5. Max Difference (A[j] - A[i])

Find maximum value of A[j] - A[i] such that j > i.

🟦 Topic 2: Search & Frequency Count (3)

6. First Repeating Element

Return the first element that repeats (based on first appearance).

7. First Non-Repeating Element

Return the first element that appears only once.

8. Unique Intersection of Two Arrays

Return all common elements (no duplicates), any order.

🟧 Topic 3: Prefix Sum, Kadane & Sliding Window (6)

9. Kadane's Algorithm

Find maximum subarray sum.

10. Max Sum in Window of Size k

Fixed-size sliding window: find max sum of any window of size k.

11. Count Subarrays with Sum = K

Count how many subarrays have exact sum equal to K.

12. Longest Subarray with Sum = K

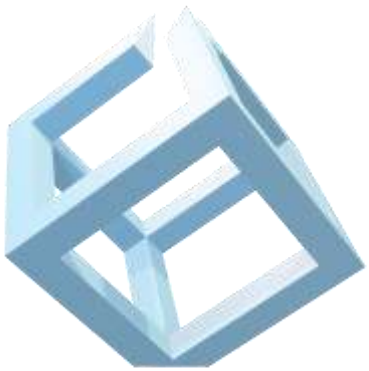
Return length of longest subarray whose sum is K.

13. Minimum Length Subarray with Sum ≥ X

Variable-size sliding window technique.

14. Even-Odd Index Prefix Sums

Build two prefix arrays: one for even indices, one for odd.



🟥 Topic 4: Two-Pointer & Rearrangement (5)

15. Pair Sum (Sorted Array)

Find indices of two numbers in a sorted array that add up to a target.

16. Triplet Sum = 0

Return YES if any triplet sums to 0.

17. Move Zeros to End

Shift all 0s to the end of the array. Keep order of non-zero elements.

18. Rearrange Alternate Signs

Rearrange array as +, –, +, –, … while maintaining original order.

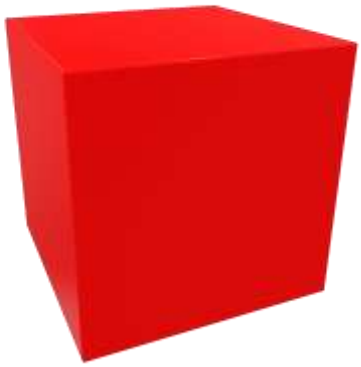
19. Stable Even-Odd Segregation

Put even numbers first, then odds, maintaining their order.

🟪 Topic 5: Product / Majority / Special Patterns (6)

20. Product of Array Except Self

Return product of all elements except the current one, no division.



21. Majority Element > N/3

Return all elements that occur more than ⌊N/3⌋ times.

22. Leaders in an Array

Element is a leader if greater than all elements to its right.

23. Alternate Sign Arrangement (Focused Version)

Another version for hands-on index placement practice.

24. Max Index Difference (j - i where A[j] ≥ A[i])

Solve in O(N) using pre-processing.

25. Merge Sorted Arrays Without Extra Space (Gap Method)

Merge two sorted arrays into one sorted sequence in-place.

🎯 Topic 6: Dutch National Flag Algorithm (2)

26. Sort 0s, 1s, 2s

Use Dutch National Flag Algorithm to sort 0, 1, 2 in O(N) time.

27. [BONUS] Sort Colors (1 to k) (skip for now)

Generalized DNF problem for values from 1 to k. (Later after hashmaps)
