Arrays

Week 1 – Basics & Easy Problems

- **Day 1**: Array intro in C++ → declaration, initialization, input/output.
  Problems: HackerRank "Arrays Introduction", "Array Reversal".
- **Day 2**: Traversal, insertion, deletion (manual shifting).
  Problems: LeetCode [27. Remove Element].
- **Day 3**: Searching (linear, binary).
  Problems: LeetCode [704. Binary Search].
- **Day 4**: Sorting basics (Bubble, Selection, Insertion).
  Problems: HackerRank "Sorting".
- **Day 5**: Two-pointer basics.
  Problems: LeetCode [977. Squares of a Sorted Array].
- **Day 6**: Prefix sums (1D arrays).
  Problems: LeetCode [303. Range Sum Query – Immutable].
- **Day 7 (Review + Practice)**: Revise week's topics.
  Problems: 5–6 mixed easy array problems.

Week 2 – Core Problem-Solving Patterns

- **Day 8**: Kadane's Algorithm (max subarray sum).
  Problems: LeetCode [53. Maximum Subarray].
- **Day 9**: Sliding window (fixed size).
  Problems: LeetCode [643. Maximum Average Subarray I].
- **Day 10**: Sliding window (variable size).
  Problems: LeetCode [209. Minimum Size Subarray Sum].
- **Day 11**: Frequency arrays & hashing basics.
  Problems: LeetCode [169. Majority Element].
- **Day 12**: Subarrays & subsets.
  Problems: LeetCode [560. Subarray Sum Equals K].
- **Day 13**: Rotations & rearrangements.
  Problems: LeetCode [189. Rotate Array].
- **Day 14 (Review + Practice)**: Solve 5–6 problems from Week 2.

Week 3 – Advanced Arrays

- **Day 15**: Matrix basics (2D arrays).
  Problems: LeetCode [73. Set Matrix Zeroes].
- **Day 16**: Matrix traversal (spiral, diagonal).
  Problems: LeetCode [54. Spiral Matrix].
- **Day 17**: Searching in 2D matrix.
  Problems: LeetCode [74. Search a 2D Matrix].
- **Day 18**: Hard problems practice (combine concepts).
  Problems: LeetCode [4. Median of Two Sorted Arrays].
- **Day 19**: Revise all concepts → prefix sums, sliding window, Kadane's.
- **Day 20**: Mock test → Pick 8–10 random array problems (mix easy/medium).

Arrays gemi

Week 1: The Bedrock – Fundamentals & Advanced Sorting

This week ensures your foundation is flawless and introduces crucial sorting algorithms.

- **Day 1: C++ Arrays & Vectors**
  - **Topics:** Static C-style arrays vs. std::vector. Declaration, initialization, memory differences (stack vs. heap). Using iterators. Common std::vector methods: push_back, pop_back, size, begin, end. LeetCode: 344. Reverse String, LeetCode 189. Rotate Arra
  - **Problems:** HackerRank "Arrays Introduction", LeetCode [1480. Running Sum of 1d Array].
- **Day 2: Basic Operations & Traversal**
  - **Topics:** Manual traversal, insertion, and deletion by shifting elements. Time complexity analysis (O(n), O(1)).
  - **Problems:** LeetCode [27. Remove Element], LeetCode [26. Remove Duplicates from Sorted Array].
- **Day 3: Searching Algorithms**

  - **Topics:** Linear Search. **Binary Search** on sorted arrays (iterative and recursive). Understanding the search space and condition low <= high.
  - **Problems:** LeetCode [704. Binary Search], LeetCode [35. Search Insert Position].
- **Day 4: Basic Sorting Algorithms**
  - **Topics:** Bubble Sort, Selection Sort, Insertion Sort. Understand why they are O(n2) and their best/worst-case scenarios.
  - **Problems:** Implement all three from scratch. HackerRank "Correctness and the Loop Invariant".
- **Day 5: Advanced Sorting Algorithms I (Merge Sort)**
  - **Topics:** The "Divide and Conquer" paradigm. How Merge Sort works. Understand its time complexity (O(nlogn)) and space complexity (O(n)).
  - **Problems:** LeetCode [912. Sort an Array] (solve it using Merge Sort).
- **Day 6: Advanced Sorting Algorithms II (Quick Sort)**
  - **Topics:** The Lomuto and Hoare partition schemes. Pivot selection. Worst-case (O(n2)) and average-case (O(nlogn)) complexity. In-place sorting.
  - **Problems:** Re-solve LeetCode [912. Sort an Array] using Quick Sort. LeetCode [215. Kth Largest Element in an Array] (using the Quickselect idea).
- **Day 7: Review & Practice**
  - **Topics:** Compare all sorting algorithms. When to use which? Practice binary search on tricky conditions.
  - **Problems:** LeetCode [34. Find First and Last Position of Element in Sorted Array], LeetCode [153. Find Minimum in Rotated Sorted Array].

Week 2: Core Problem-Solving Patterns

This week focuses on the essential patterns that solve the majority of interview problems.

- **Day 8: Two Pointers**
  - **Topics:** Converging pointers, fast/slow pointers. Using them for searching pairs, reversing, and palindrome checks.
  - **Problems:** LeetCode [977. Squares of a Sorted Array], LeetCode [11. Container With Most Water].

- **Day 9: Sliding Window (Fixed Size)**
    - **Topics:** The basic sliding window concept. Maintaining a window of size 'k' by adding an element to the end and removing one from the start.
    - **Problems:** LeetCode [643. Maximum Average Subarray I], LeetCode [219. Contains Duplicate II].
- **Day 10: Sliding Window (Variable Size)**
    - **Topics:** Expanding and shrinking the window based on conditions to find the longest/shortest subarray that satisfies a property.
    - **Problems:** LeetCode [209. Minimum Size Subarray Sum], LeetCode [3. Longest Substring Without Repeating Characters].
- **Day 11: Prefix & Suffix Sums**
    - **Topics:** Pre-computation to answer range queries in O(1). 1D prefix sums, suffix sums, and combining them.
    - **Problems:** LeetCode [303. Range Sum Query – Immutable], LeetCode [238. Product of Array Except Self].
- **Day 12: Kadane's Algorithm & Variations**
    - **Topics:** Finding the maximum/minimum sum subarray. Understanding the logic of discarding negative-sum prefixes.
    - **Problems:** LeetCode [53. Maximum Subarray], LeetCode [918. Maximum Sum Circular Subarray].
- **Day 13: Hashing & Frequency Counting**
    - **Topics:** Using std::unordered_map or frequency arrays (int arr[26]) to count occurrences and solve problems efficiently.
    - **Problems:** LeetCode [1. Two Sum], LeetCode [169. Majority Element].
- **Day 14: Review & Pattern Recognition**
    - **Topics:** Practice recognizing which pattern applies to a given problem.
    - **Problems:** LeetCode [560. Subarray Sum Equals K], LeetCode [121. Best Time to Buy and Sell Stock], LeetCode [438. Find All Anagrams in a String].

Week 3: 2D Arrays & Dynamic Programming

Dive into matrices and the powerful technique of DP on arrays.

- **Day 15: 2D Array (Matrix) Basics**
  - **Topics:** Declaration in C++ (vector<vector<int>>). Traversal (row-major, column-major). Passing matrices to functions.
  - **Problems:** LeetCode [867. Transpose Matrix], LeetCode [73. Set Matrix Zeroes].
- **Day 16: Matrix Traversals**
  - **Topics:** Spiral, diagonal, and boundary traversals. Thinking about layer-by-layer or direction-based logic.
  - **Problems:** LeetCode [54. Spiral Matrix], LeetCode [498. Diagonal Traverse].
- **Day 17: Searching in a 2D Matrix**
  - **Topics:** Applying binary search on sorted matrices. Two main types: sorted rows/columns, and fully sorted.
  - **Problems:** LeetCode [74. Search a 2D Matrix], LeetCode [240. Search a 2D Matrix II].
- **Day 18: Array Rotations & Rearrangements**
  - **Topics:** In-place rotations (reversal algorithm). Rearranging elements based on properties (e.g., positive/negative). The Dutch National Flag problem.
  - **Problems:** LeetCode [189. Rotate Array], LeetCode [75. Sort Colors].
- **Day 19: Intro to Dynamic Programming (DP) on Arrays**
  - **Topics:** The concept of optimal substructure and overlapping subproblems. Understanding the state transition. 1D DP.
  - **Problems:** LeetCode [70. Climbing Stairs], LeetCode [198. House Robber].
- **Day 20: More 1D DP on Arrays**
  - **Topics:** Building on the previous day with more complex state logic.
  - **Problems:** LeetCode [322. Coin Change], LeetCode [300. Longest Increasing Subsequence].
- **Day 21: Review & Medium/Hard Practice**
  - **Topics:** Combine matrix logic with other patterns.
  - **Problems:** LeetCode [48. Rotate Image], LeetCode [5. Longest Palindromic Substring] (DP solution).

Week 4: Advanced Combinations & Mastery

This final week covers complex interactions with other data structures and niche topics to round out your knowledge.

- **Day 22: Arrays & Stacks**
    - **Topics:** Using a monotonic stack to find the next/previous greater/smaller element.
    - **Problems:** LeetCode [739. Daily Temperatures], LeetCode [84. Largest Rectangle in Histogram].
- **Day 23: Arrays & Heaps (Priority Queues)**
    - **Topics:** Using heaps to solve problems involving the 'Kth' element, merging, or scheduling.
    - **Problems:** LeetCode [973. K Closest Points to Origin], LeetCode [253. Meeting Rooms II].
- **Day 24: Bit Manipulation on Arrays**
    - **Topics:** Using XOR to find unique/missing elements. Using bitmasks to handle subsets.
    - **Problems:** LeetCode [136. Single Number], LeetCode [268. Missing Number].
- **Day 25: Very Hard Problems (Combining Concepts)**
    - **Topics:** Tackle problems that require multiple patterns (e.g., binary search on the answer + sliding window).
    - **Problems:** LeetCode [4. Median of Two Sorted Arrays], LeetCode [42. Trapping Rain Water].
- **Day 26: Linear Time Sorting & Special Cases**
    - **Topics:** Counting Sort, Radix Sort. Understand their constraints and when they are superior to comparison-based sorts.
    - **Problems:** Implement Counting Sort from scratch.
- **Day 27: Mock Interview Session 1**
    - **Goal:** Pick 3-4 random array problems (1 Easy, 2 Medium, 1 Hard) and solve them under a time limit (e.g., 90 minutes). Simulate an interview by explaining your approach out loud.
- **Day 28: Mock Interview Session 2 & Final Review**

- **Goal:** Repeat the mock interview process. Review any topic from the last 28 days where you felt weak. Congratulate yourself on your hard work! 🏆