Array Algorithm Patterns - Cheat Sheet

Two Pointers

Used on sorted arrays or when comparing elements from both ends. E.g., Pair with a given sum, remove duplicates, reverse array.

Sliding Window

Used for subarray problems like max/min sum or length. E.g., Longest substring, min size subarray.

Prefix Sum / Cumulative Sum

Quick sum of subarrays. E.g., Subarray sum equals k.

Binary Search

Efficient searching in sorted arrays. E.g., Search in rotated array.

Sorting + Greedy

Optimize by sorting and making greedy choices. E.g., Merge intervals.

Hash Map / Frequency Counter

Track occurrences. E.g., Two sum, majority element.

Kadanes Algorithm

Find max subarray sum. E.g., Max subarray.

Divide and Conquer

Break problem into subproblems. E.g., Merge sort, max subarray.

Backtracking

Explore all permutations/combinations. E.g., Subsets, permutations.

Monotonic Stack / Deque

Maintain order, find next greater. E.g., Daily temperatures.

In-Place Rearrangement

Reorder without extra space. E.g., Move zeros, cyclic sort.

Matrix as 2D Array

Array Algorithm Patterns - Cheat Sheet

Grid-based problems. E.g., Rotate matrix, spiral order.

Difference Arrays

Efficient range updates. E.g., Range addition.

2D Prefix Sums

Fast rectangular submatrix queries. E.g., Sum of submatrix.

Reservoir Sampling

Random index from stream. E.g., Random pick index.

Partitioning Arrays

Divide into balanced subarrays. E.g., Partition equal subset.

Heap with Arrays

Get min/max efficiently. E.g., Kth largest, merge k arrays.

Disjoint Set Union

Group connected components. E.g., Number of provinces.

Meet in the Middle

Split problem for optimization. E.g., Subset sum.

Mos Algorithm

Efficient offline queries. E.g., Count distinct elements.

Sparse Tables

Fast immutable range queries. E.g., Range minimum query.

Digit DP

Count numbers with constraints. E.g., Unique digits count.

Convex Hull Trick

Optimize linear functions. Used in CP.

Array Algorithm Patterns - Cheat Sheet

Suffix Arrays + LCP

String array hybrid for substring problems.

Implicit Arrays

Use index tricks for simulation. E.g., Missing number.

Mathematical Arrays

Use math properties to solve. E.g., Sum formulas.

Greedy with Arrays

Apply local optimal choice strategy. E.g., Jump Game, Gas Station.

Cycle Sort

Used when elements are in the range 1 to N. E.g., Find missing number.

Counting Sort

Non-comparison sorting for integers. Used in radix sort.

Radix Sort

Efficient non-comparison sort for fixed-length numbers.

Bucket Sort

Divide elements into buckets and sort each. E.g., Top K Frequent Elements.

Topological Sort on Arrays

Used when arrays represent graph edges. E.g., Course Schedule.

Multi-Dimensional DP on Arrays

Used in problems with multiple states. E.g., 0/1 Knapsack.

Greedy + Sorting with Custom Comparator

Sort with rules. E.g., Largest Number, Reorder Data.

Difference Pair/Triplet in Array

Find pairs with fixed difference. E.g., K-diff Pairs.

Array Algorithm Patterns - Cheat Sheet

Prefix XOR Array

Used in XOR subarray problems. E.g., Count of XORs with given value.

Sliding Window Max with Deque

Efficient max/min of each subarray. E.g., Sliding Window Maximum.

Merge Sort Inversion Count

Count how far array is from being sorted.

Balanced Partition Using DP

Partition array into equal sum sets.

Range Sum Query using Segment Tree

Update/query range in O(log n).

Lazy Propagation in Segment Trees

Optimize range updates.