# Arrays & Hashing – Core Patterns Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Pattern | Core Idea | Mental Idea | Example Problem |
| 1 | Frequency Counting | Count how many times each element occurs | "I need to know how many times something appears." | Valid Anagram, Top K Frequent Elements |
| 2 | Hash Map Lookup (Complement) | Find complements using a hash map | "Have I already seen something that pairs with this?" | Two Sum, Pair With Given Difference |
| 3 | Prefix Sum | Use running totals to find subarrays | "I need something between i and j — can I reuse prefix?" | Subarray Sum Equals K, Longest Subarray Sum = K |
| 4 | Set Membership / Uniqueness | Check for duplicates or existence | "I only care about whether it exists or not." | Contains Duplicate, Longest Consecutive Sequence |
| 5 | Sorting + Hashing | Normalize data before grouping/hashing | "If I sort or normalize, identical ones look the same." | Group Anagrams, Merge Intervals |
| 6 | Index Mapping / Positional Hashing | Store earliest/last index of value/prefix | "I need to remember where something first appeared." | Longest Subarray Sum = K, First Unique Character |
| 7 | Bucket Counting / Frequency Bucketing | Group items by frequency | "I have counts — can I access items by how often they appear?" | Top K Frequent, Sort Characters by Frequency |
| 8 | Modular Hashing / Math Hashing | Use modulo to handle circular/divisible sums | "I can reduce large or repeating values with mod math." | Subarray Sum Divisible by K, Continuous Subarray Sum |