LeetCode Questions for Developers: Array, Hash Map, String, and Other Patterns

Array

Array problems are critical for MERN developers for tasks like React state management, API data processing, and array manipulation in JavaScript.

1. **1**: Two Sum (Easy) - Find indices of two numbers that add up to a target.
2. **26**: Remove Duplicates from Sorted Array (Easy) - Remove duplicates in-place.
3. **27**: Remove Element (Easy) - Remove all occurrences of a value in-place.
4. **35**: Search Insert Position (Easy) - Find insertion point in a sorted array.
5. **66**: Plus One (Easy) - Increment a number represented as an array.
6. **88**: Merge Sorted Array (Easy) - Merge two sorted arrays in-place.
7. **118**: Pascal's Triangle (Easy) - Generate Pascal's triangle rows.
8. **119**: Pascal's Triangle II (Easy) - Generate a specific row of Pascal's triangle.
9. **121**: Best Time to Buy and Sell Stock (Easy) - Maximize profit from one transaction.
10. **136**: Single Number (Easy) - Find the number that appears once.
11. **169**: Majority Element (Easy) - Find element appearing more than n/2 times.
12. **189**: Rotate Array (Medium) - Rotate array right by k steps.
13. **209**: Minimum Size Subarray Sum (Medium) - Find smallest subarray with sum >= target.
14. **217**: Contains Duplicate (Easy) - Check if array contains duplicates.
15. **228**: Summary Ranges (Easy) - Find continuous ranges in a sorted array.
16. **268**: Missing Number (Easy) - Find missing number in range [0,n].
17. **283**: Move Zeroes (Easy) - Move all zeros to the end of the array.
18. **350**: Intersection of Two Arrays II (Easy) - Find intersection of two arrays.
19. **414**: Third Maximum Number (Easy) - Find third distinct maximum.
20. **448**: Find All Numbers Disappeared in an Array (Easy) - Find missing numbers.
21. **485**: Max Consecutive Ones (Easy) - Find maximum number of consecutive 1s.
22. **509**: Fibonacci Number (Easy) - Compute nth Fibonacci number using array.
23. **561**: Array Partition I (Easy) - Maximize sum of min of paired elements.
24. **566**: Reshape the Matrix (Easy) - Reshape matrix into new shape.
25. **575**: Contains Duplicate (Easy) - Check for duplicates using set.
26. **605**: Can Place Flowers (Easy) - Check if flowers can be planted without adjacency.
27. **628**: Maximum Product of Three Numbers (Easy) - Find max product of three numbers.

28. **643**: Maximum Average Subarray I (Easy) - Find max average of k-length subarray.
29. **674**: Longest Continuous Increasing Subsequence (Easy) - Find longest increasing subarray.
30. **697**: Degree of an Array (Easy) - Find shortest subarray with max frequency.
31. **724**: Find Pivot Index (Easy) - Find index where left and right sums are equal.
32. **747**: Largest Number At Least Twice of Others (Easy) - Check if largest is twice others.
33. **766**: Toeplitz Matrix (Easy) - Check if matrix is Toeplitz.
34. **832**: Flipping an Image (Easy) - Flip and invert a binary image.
35. **896**: Monotonic Array (Easy) - Check if array is monotonic.
36. **905**: Sort Array By Parity (Easy) - Move even numbers to the front.
37. **941**: Valid Mountain Array (Easy) - Check if array forms a mountain.
38. **977**: Squares of a Sorted Array (Easy) - Return sorted array of squared elements.
39. **1089**: Duplicate Zeros (Easy) - Duplicate zeros in-place.
40. **1122**: Relative Sort Array (Easy) - Sort array based on another array's order.
41. **1295**: Find Numbers with Even Number of Digits (Easy) - Count numbers with even digits.
42. **1299**: Replace Elements with Greatest Element on Right Side (Easy) - Replace with max on right.
43. **1346**: Check If N and Its Double Exist (Easy) - Check if number and its double exist.
44. **1351**: Count Negative Numbers in a Sorted Matrix (Easy) - Count negatives in sorted matrix.
45. **1365**: How Many Numbers Are Smaller Than the Current Number (Easy) - Count smaller numbers.
46. **1389**: Create Target Array in the Given Order (Easy) - Create array with given indices.
47. **1464**: Maximum Product of Two Elements in an Array (Easy) - Find max product of two numbers.
48. **1470**: Shuffle the Array (Easy) - Shuffle array in specific pattern.
49. **1480**: Running Sum of 1d Array (Easy) - Compute running sum of array.
50. **1512**: Number of Good Pairs (Easy) - Count pairs with equal elements.
51. **1588**: Sum of All Odd Length Subarrays (Easy) - Sum all odd-length subarrays.
52. **1672**: Richest Customer Wealth (Easy) - Find maximum wealth from accounts.
53. **1920**: Build Array from Permutation (Easy) - Build array using given indices.
54. **1929**: Concatenation of Array (Easy) - Concatenate array with itself.

55. **2016**: Maximum Difference Between Increasing Elements (Easy) - Find max difference.
56. **2341**: Maximum Number of Pairs in Array (Easy) - Form pairs to maximize count.
57. **2418**: Sort the People (Easy) - Sort by height.
58. **2535**: Difference Between Element Sum and Digit Sum of an Array (Easy) - Compute sum difference.
59. **2574**: Left and Right Sum Differences (Easy) - Compute absolute differences of sums.
60. **2974**: Minimum Number Game (Easy) - Minimize array through operations.

Hash Map

Hash Map problems optimize lookups and frequency counting, key for Node.js API performance and MongoDB query processing in MERN applications.

1. **1**: Two Sum (Easy) - Use hash map to find indices summing to target.
2. **136**: Single Number (Easy) - Find unique element using XOR or hash map.
3. **169**: Majority Element (Easy) - Find majority element using hash map.
4. **205**: Isomorphic Strings (Easy) - Check if strings are isomorphic using mapping.
5. **217**: Contains Duplicate (Easy) - Check for duplicates using hash set.
6. **219**: Contains Duplicate II (Easy) - Check duplicates within k distance.
7. **242**: Valid Anagram (Easy) - Check if two strings are anagrams.
8. **290**: Word Pattern (Easy) - Check if string follows a pattern.
9. **349**: Intersection of Two Arrays (Easy) - Find intersection using hash set.
10. **350**: Intersection of Two Arrays II (Easy) - Find intersection with frequency.
11. **387**: First Unique Character in a String (Easy) - Find first non-repeating character.
12. **389**: Find the Difference (Easy) - Find extra character in string.
13. **409**: Longest Palindrome (Easy) - Build longest palindrome from characters.
14. **447**: Number of Boomerangs (Medium) - Count tuples with equal distances.
15. **448**: Find All Numbers Disappeared in an Array (Easy) - Find missing numbers.
16. **454**: 4Sum II (Medium) - Count quadruplets summing to zero.
17. **496**: Next Greater Element I (Easy) - Find next greater element.
18. **532**: K-diff Pairs in an Array (Medium) - Find pairs with k difference.
19. **560**: Subarray Sum Equals K (Medium) - Find subarrays with sum k.
20. **575**: Contains Duplicate III (Medium) - Check duplicates within k and t.
21. **594**: Longest Harmonious Subsequence (Easy) - Find longest subsequence with diff 1.
22. **599**: Minimum Index Sum of Two Lists (Easy) - Find common restaurants.

23. **645**: Set Mismatch (Easy) - Find duplicate and missing number.
24. **697**: Degree of an Array (Easy) - Find shortest subarray with max frequency.
25. **705**: Design HashSet (Easy) - Implement a hash set.
26. **706**: Design HashMap (Easy) - Implement a hash map.
27. **771**: Jewels and Stones (Easy) - Count jewels in stones.
28. **811**: Subdomain Visit Count (Easy) - Count subdomain visits.
29. **888**: Fair Candy Swap (Easy) - Find pair to equalize candy sums.
30. **961**: N-Repeated Element in Size 2N Array (Easy) - Find element repeated N times.
31. **966**: Vowel Spellchecker (Medium) - Implement spellchecker with rules.
32. **1002**: Find Common Characters (Easy) - Find common characters across words.
33. **1207**: Unique Number of Occurrences (Easy) - Check if frequency counts are unique.
34. **1331**: Rank Transform of an Array (Easy) - Replace elements with their ranks.
35. **1370**: Increasing Decreasing String (Easy) - Sort string by frequency.
36. **1460**: Make Two Arrays Equal by Reversing Sub-arrays (Easy) - Check if arrays can be equal.
37. **1512**: Number of Good Pairs (Easy) - Count identical pairs.
38. **1590**: Make Sum Divisible by P (Medium) - Find subarray to remove.
39. **1636**: Sort Array by Increasing Frequency (Easy) - Sort by frequency.
40. **1748**: Sum of Unique Elements (Easy) - Sum elements appearing once.
41. **1832**: Check if All Characters Have Equal Number of Occurrences (Easy) - Check equal frequencies.
42. **1941**: Check if All Characters Have Equal Number of Occurrences (Easy) - Verify character frequencies.
43. **2022**: Convert 1D Array Into 2D Array (Easy) - Reshape array into matrix.
44. **2085**: Count Common Words With One Occurrence (Easy) - Count words appearing once.
45. **2156**: Find Substring With Given Hash Value (Hard) - Find substring with hash.
46. **2287**: Rearrange Characters to Make Target String (Easy) - Check if target can be formed.
47. **2357**: Make Array Zero by Subtracting Equal Amounts (Easy) - Minimize operations.
48. **2367**: Number of Arithmetic Triplets (Easy) - Count triplets with difference.
49. **2418**: Sort the People (Easy) - Sort by height using frequency map.
50. **2441**: Largest Positive Integer That Exists With Its Negative (Easy) - Find max k with -k.

51. **2351**: First Letter to Appear Twice (Easy) - Find first repeated character.
52. **3162**: Find the Number of Good Pairs I (Easy) - Count pairs with given condition.

String

String problems are essential for MERN developers for tasks like input validation, API response formatting, and React UI string rendering.

1. **3**: Longest Substring Without Repeating Characters (Medium) - Find longest unique substring.
2. **5**: Longest Palindromic Substring (Medium) - Find longest palindrome.
3. **14**: Longest Common Prefix (Easy) - Find common prefix in strings.
4. **20**: Valid Parentheses (Easy) - Check if parentheses are balanced.
5. **28**: Implement strStr() (Easy) - Find substring index.
6. **38**: Count and Say (Easy) - Generate nth term of count-and-say sequence.
7. **58**: Length of Last Word (Easy) - Find length of last word.
8. **67**: Add Binary (Easy) - Add two binary strings.
9. **125**: Valid Palindrome (Easy) - Check if string is palindrome.
10. **151**: Reverse Words in a String (Medium) - Reverse words in a string.
11. **205**: Isomorphic Strings (Easy) - Check if strings are isomorphic.
12. **242**: Valid Anagram (Easy) - Check if strings are anagrams.
13. **344**: Reverse String (Easy) - Reverse a string in-place.
14. **345**: Reverse Vowels of a String (Easy) - Reverse only vowels.
15. **383**: Ransom Note (Easy) - Check if ransom note can be constructed.
16. **387**: First Unique Character in a String (Easy) - Find first non-repeating character.
17. **389**: Find the Difference (Easy) - Find extra character.
18. **392**: Is Subsequence (Easy) - Check if string is a subsequence.
19. **409**: Longest Palindrome (Easy) - Build longest palindrome.
20. **415**: Add Strings (Easy) - Add two numbers as strings.
21. **434**: Number of Segments in a String (Easy) - Count segments.
22. **438**: Find All Anagrams in a String (Medium) - Find all anagram substrings.
23. **459**: Repeated Substring Pattern (Easy) - Check if string is repeated pattern.
24. **520**: Detect Capital (Easy) - Check valid capitalization.
25. **521**: Longest Uncommon Subsequence I (Easy) - Find longest uncommon subsequence.
26. **541**: Reverse String II (Easy) - Reverse every k characters.

27. **551**: Student Attendance Record I (Easy) - Check valid attendance record.
28. **557**: Reverse Words in a String III (Easy) - Reverse each word.
29. **606**: Construct String from Binary Tree (Easy) - Create string from tree.
30. **647**: Palindromic Substrings (Medium) - Count palindromic substrings.
31. **657**: Robot Return to Origin (Easy) - Check if robot returns to origin.
32. **680**: Valid Palindrome II (Easy) - Check palindrome with one deletion.
33. **696**: Count Binary Substrings (Easy) - Count binary substring patterns.
34. **709**: To Lower Case (Easy) - Convert string to lowercase.
35. **763**: Partition Labels (Medium) - Partition string into labels.
36. **771**: Jewels and Stones (Easy) - Count jewels in stones.
37. **796**: Rotate String (Easy) - Check if string is a rotation.
38. **819**: Most Common Word (Easy) - Find most frequent non-banned word.
39. **824**: Goat Latin (Easy) - Convert string to Goat Latin.
40. **830**: Positions of Large Groups (Easy) - Find large character groups.
41. **859**: Buddy Strings (Easy) - Check if strings are buddy strings.
42. **893**: Groups of Special-Equivalent Strings (Medium) - Group equivalent strings.
43. **917**: Reverse Only Letters (Easy) - Reverse only letters in string.
44. **925**: Long Pressed Name (Easy) - Check if name is long pressed.
45. **929**: Unique Email Addresses (Easy) - Count unique email addresses.
46. **937**: Reorder Data in Log Files (Easy) - Reorder log entries.
47. **953**: Verifying an Alien Dictionary (Easy) - Check if words follow alien order.
48. **966**: Vowel Spellchecker (Medium) - Implement spellchecker.
49. **1047**: Remove All Adjacent Duplicates In String (Easy) - Remove adjacent duplicates.
50. **1108**: Defanging an IP Address (Easy) - Replace dots with [.] in IP address.
51. **1903**: Largest Odd Number in String (Easy) - Find largest odd number substring.
52. **2185**: Counting Words With a Given Prefix (Easy) - Count words with prefix.
53. **2264**: Largest 3-Same-Digit Number in String (Easy) - Find largest three same digits.
54. **2283**: Check if Number Has Equal Digit Count and Digit Value (Easy) - Verify digit counts.
55. **2942**: Find Words Containing Character (Easy) - Find words with specific character.

Other

This category includes problems that don't clearly fit Array, Hash Map, or String patterns, covering patterns like Dynamic Programming, Greedy, Graph, Tree, Math, or invalid/ambiguous problems.

1. **70**: Climbing Stairs (Easy) - Dynamic Programming: Count ways to climb stairs.
2. **202**: Happy Number (Easy) - Math/Hash Set: Check if a number is happy.
3. **234**: Palindrome Linked List (Easy) - Linked List: Check if linked list is palindrome.
4. **704**: Binary Search (Easy) - Binary Search: Find target in sorted array.
5. **904**: Fruit Into Baskets (Medium) - Sliding Window: Collect max fruits with two types.
6. **1662**: Check If Two String Arrays are Equivalent (Easy) - String: Concatenate and compare strings.
7. **1816**: Truncate Sentence (Easy) - String: Truncate sentence to k words.
8. **1979**: Find Greatest Common Divisor of Array (Easy) - Math: Find GCD of array elements.
9. **2078**: Two Furthest Houses With Different Colors (Easy) - Greedy: Maximize distance between colors.
10. **2154**: Keep Multiplying Found Values by Two (Easy) - Array: Multiply elements by 2.
11. **2162**: Minimum Cost of a Path With Special Roads (Medium) - Graph: Find min cost path.
12. **2180**: Count Integers With Even Digit Sum (Easy) - Math: Count numbers with even digit sum.
13. **2413**: Smallest Even Multiple (Easy) - Math: Find smallest multiple of 2 and n.
14. **2496**: Maximum Value of a String in an Array (Easy) - String: Max value of string in array.
15. **2540**: Minimum Common Value (Easy) - Set/Two-pointer: Find common element.
16. **2544**: Alternating Digit Sum (Easy) - Math: Compute alternating digit sum.
17. **2869**: Minimum Operations to Collect Elements (Easy) - Greedy: Collect k elements.
18. **2894**: Divisible and Non-divisible Sums Difference (Easy) - Math: Compute sum difference.
19. **3069**: Distribute Elements Into Two Arrays I (Easy) - Greedy: Distribute into two arrays.
20. **3142**: Check if Grid Satisfies Conditions (Easy) - Matrix: Check grid conditions.
21. **3190**: Find Minimum Operations to Make All Elements Divisible by Three (Easy) - Math: Minimize operations.