Array problem

Array Problems - Practice Questions

A curated list of array manipulation problems for coding practice and interviews.

---

1. Second Largest Element

Problem Statement:

Find the second largest element in an array. If the second largest element doesn't exist, return -1.

Example:

```
Input: [12, 35, 1, 10, 34, 1]
Output: 34

Input: [10, 10, 10]
Output: -1
```

Constraints:

- Array size: 0 ≤ n ≤ 10^5
- Element values: -10^9 ≤ arr[i] ≤ 10^9

---

2. Third Largest Element

Problem Statement:

Find the third largest distinct element in an array. If it doesn't exist, return -1.

Example:

```
Input: [10, 4, 3, 50, 23, 90]
Output: 23

Input: [10, 10]
Output: -1
```

Constraints:

- Array size: 0 ≤ n ≤ 10^5
- Element values: -10^9 ≤ arr[i] ≤ 10^9

3. Reverse an Array

Problem Statement:

Reverse the elements of an array in-place.

Example:

```
Input: [1, 2, 3, 4, 5]
Output: [5, 4, 3, 2, 1]

Input: [10]
Output: [10]
```

Constraints:

- Array size: 0 ≤ n ≤ 10^5
- Must be done in-place with O(1) extra space

4. Reverse Array in Groups

Problem Statement:

Given an array and a group size k, reverse every group of k consecutive elements. If the number of elements is not a multiple of k, reverse the remaining elements as well.

Example:

```
Input: arr = [1, 2, 3, 4, 5, 6, 7, 8], k = 3
Output: [3, 2, 1, 6, 5, 4, 8, 7]

Input: arr = [1, 2, 3, 4, 5], k = 3
Output: [3, 2, 1, 5, 4]
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Group size: 1 ≤ k ≤ n

---

5. Rotate Array

Problem Statement:

Rotate an array to the right by k positions.

Example:

```
Input: arr = [1, 2, 3, 4, 5, 6, 7], k = 3
Output: [5, 6, 7, 1, 2, 3, 4]

Input: arr = [-1, -100, 3, 99], k = 2
Output: [3, 99, -1, -100]
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Rotation count: 0 ≤ k ≤ 10^9
- Must be done in-place with O(1) extra space

---

6. Three Great Candidates

Problem Statement:

Given an array of integers representing candidate scores, find the three candidates with the highest scores and return them in descending order.

Example:

```
Input: [10, 4, 3, 50, 23, 90]
Output: [90, 50, 23]

Input: [1, 2]
Output: [2, 1]
```

Constraints:

- Array size: 0 ≤ n ≤ 10^5
- Element values: -10^9 ≤ scores[i] ≤ 10^9

7. Max Consecutive Ones

Problem Statement:

Given a binary array (containing only 0s and 1s), find the maximum number of consecutive 1s.

Example:

```
Input: [1, 1, 0, 1, 1, 1, 0, 1, 1]
Output: 3

Input: [1, 0, 1, 0, 1]
Output: 1
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Array contains only 0s and 1s

8. Move All Zeroes To End

Problem Statement:

Move all zeros in an array to the end while maintaining the relative order of non-zero elements.

Example:

```
Input: [0, 1, 0, 3, 12]
Output: [1, 3, 12, 0, 0]

Input: [0, 0, 1]
Output: [1, 0, 0]
```

Constraints:

- Array size: 1 ≤ n ≤ 10^4
- Must be done in-place

---

9. Wave Array

Problem Statement:

Rearrange array elements in a wave-like pattern where arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= arr[5]...

Example:

```
Input: [10, 5, 6, 3, 2, 20, 100, 80]
Output: [10, 5, 6, 2, 20, 3, 100, 80] (one possible answer)

Input: [20, 10, 8, 6, 4, 2]
Output: [20, 8, 10, 4, 6, 2] (one possible answer)
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Multiple valid answers possible

---

10. Plus One

Problem Statement:

Given an array representing a non-negative integer where each element is a digit (0-9), add one to the integer represented by the array.

Example:

```
Input: [1, 2, 3]  (represents 123)
Output: [1, 2, 4]  (represents 124)

Input: [9, 9, 9]  (represents 999)
Output: [1, 0, 0, 0]  (represents 1000)

Input: [0]
Output: [1]
```

Constraints:

- Array size: 1 ≤ n ≤ 100
- Each element: 0 ≤ digits[i] ≤ 9
- No leading zeros except for the number 0 itself

11. Stock Buy and Sell – One Transaction

Problem Statement:

You are given an array where each element represents the price of a stock on a given day. Find the maximum profit you can achieve by buying and selling the stock once. You must buy before you sell.

Example:

```
Input: [7, 1, 5, 3, 6, 4]
Output: 5  (buy at 1, sell at 6)
```

```
Input: [7, 6, 4, 3, 1]
Output: 0  (no profit possible)
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Price values: 0 ≤ prices[i] ≤ 10^4

12. Stock Buy and Sell – Multiple Transactions

Problem Statement:

You are given an array where each element represents the price of a stock on a given day. Find the maximum profit you can achieve by completing as many transactions as you like (buying and selling multiple times). You cannot hold multiple stocks at once (must sell before buying again).

Example:

```
Input: [7, 1, 5, 3, 6, 4]
Output: 7  (buy at 1, sell at 5, profit = 4; buy at 3, sell at 6, profit = 3)

Input: [1, 2, 3, 4, 5]
Output: 4  (buy at 1, sell at 5)
```

Constraints:

- Array size: 1 ≤ n ≤ 3 × 10^4
- Price values: 0 ≤ prices[i] ≤ 10^4

13. Remove Duplicates from Sorted Array

Problem Statement:

Remove duplicates from a sorted array in-place and return the new length. The relative order of elements should be maintained.

Example:

```
Input: [1, 1, 2, 2, 3, 4, 4]
Output: 4, array becomes [1, 2, 3, 4, _, _, _]

Input: [1, 1, 1]
Output: 1, array becomes [1, _, _]
```

Constraints:

- Array size: 0 ≤ n ≤ 3 × 10^4
- Array is sorted in non-decreasing order
- Element values: -100 ≤ arr[i] ≤ 100

14. Alternate Positive Negative

Problem Statement:

Rearrange array elements such that positive and negative numbers appear alternately. The order of appearance should be maintained. If there are more positive or negative numbers, they should appear at the end.

Example:

```
Input: [1, 2, 3, -4, -1, 4]
Output: [-4, 1, -1, 2, 3, 4] or [1, -4, 2, -1, 3, 4]

Input: [5, -2, 5, 2, 4, 7, 1, 8, 0, -8]
Output: [-2, 5, -8, 5, 2, 4, 7, 1, 8, 0] (one possible answer)
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Element values: -10^9 ≤ arr[i] ≤ 10^9

15. Array Leaders

Problem Statement:

An element is a leader if it is greater than or equal to all elements to its right side. The rightmost element is always a leader. Find all leaders in the array.

Example:

```
Input: [16, 17, 4, 3, 5, 2]
Output: [17, 5, 2]

Input: [1, 2, 3, 4, 5]
Output: [5]

Input: [5, 4, 3, 2, 1]
Output: [5, 4, 3, 2, 1]
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Element values: 0 ≤ arr[i] ≤ 10^7

---

16. Missing and Repeating in Array

Problem Statement:

Given an array of size n containing numbers from 1 to n, one number is missing and one number is repeating. Find both the missing and repeating numbers.

Example:

```
Input: [4, 3, 6, 2, 1, 1]
Output: Repeating = 1, Missing = 5

Input: [2, 2]
Output: Repeating = 2, Missing = 1
```

Constraints:

- Array size: 2 ≤ n ≤ 10^5
- Array contains numbers from 1 to n

- Exactly one number is missing and one is repeating

17. Missing Ranges of Numbers

Problem Statement:

Given a sorted array of distinct integers and a range [lower, upper], return all missing ranges between lower and upper that are not present in the array.

Example:

```
Input: arr = [0, 1, 3, 50, 75], lower = 0, upper = 99
Output: ["2", "4→49", "51→74", "76→99"]

Input: arr = [], lower = 1, upper = 1
Output: ["1"]

Input: arr = [-1], lower = -1, upper = -1
Output: []
```

Constraints:

- Array size: 0 ≤ n ≤ 100
- 10^9 ≤ lower ≤ upper ≤ 10^9
- All array elements are within [lower, upper]
- Array is sorted and contains distinct integers

18. Sum of all Subarrays

Problem Statement:

Find the sum of all possible subarrays of a given array. A subarray is a contiguous part of the array.

Example:

```
Input: [1, 2, 3]
Subarrays: [1], [2], [3], [1,2], [2,3], [1,2,3]
```

```
Sum: 1 + 2 + 3 + 3 + 5 + 6 = 20
Output: 20

Input: [1, -2, 3]
Output: 8
```

Constraints:

- Array size: 1 ≤ n ≤ 10^5
- Element values: -10^4 ≤ arr[i] ≤ 10^4

Difficulty Levels

Easy

1. Second Largest Element
2. Third Largest Element
3. Reverse an Array
4. Max Consecutive Ones
5. Move All Zeroes To End
6. Remove Duplicates from Sorted Array

Medium

1. Reverse Array in Groups
2. Rotate Array
3. Three Great Candidates
4. Wave Array
5. Plus One
6. Stock Buy and Sell – One Transaction
7. Stock Buy and Sell – Multiple Transactions
8. Alternate Positive Negative

9. Array Leaders

Hard

1. Missing and Repeating in Array
2. Missing Ranges of Numbers
3. Sum of all Subarrays

Tips for Solving

1. **Read Carefully**: Understand the problem constraints and examples
2. **Think of Edge Cases**: Empty arrays, single elements, all same elements
3. **Start Simple**: First solve with a brute force approach
4. **Optimize**: Then think about time and space complexity improvements
5. **Test Thoroughly**: Verify your solution with multiple test cases
6. **Practice Regularly**: Consistency is key to mastering array problems

---

Good Luck with Your Practice! 🚀