

## Day 2

1. Sort array by parity **[Easy]** **[Capital One, Facebook, Google]**

<https://interviewprep.appliedroots.com/lecture/2/interview-preparation-course/1304/sort-array-by-parity-problem-statement-leetcode/18/module-5-problem-solving>

2. Create Target Array in the Given order **[Easy]** **[Apple]**

<https://interviewprep.appliedroots.com/lecture/2/interview-preparation-course/1302/create-target-array-in-the-given-order-problem-statementleetcode/18/module-5-problem-solving>

3. Replace elements with Greatest Element on Right Side **[Easy]** **[Amazon, Adobe]**

<https://interviewprep.appliedroots.com/lecture/2/interview-preparation-course/1307/replace-elements-with-greatest-element-on-right-side-problem-statement-leetcode/18/module-5-problem-solving>

4. Shortest Unsorted Continuous Subarray **[Medium]** **[Amazon]**

<https://interviewprep.appliedroots.com/lecture/2/interview-preparation-course/1394/shortest-unsorted-continuous-subarray-problem-statement-leetcode/18/module-5-problem-solving>

5. Find leaders in an array **[Easy]** **[Amazon, Payu]**

<https://interviewprep.appliedroots.com/lecture/2/interview-preparation-course/1018/find-leaders-in-an-array/18/module-5-problem-solving>

### Practice Questions

1. Given an integer array nums of length n where all the integers of nums are in the range [1, n] and each integer appears once or twice, return an array of all the integers that appears twice.

You must write an algorithm that runs in  $O(n)$  time and uses only constant extra space. **[Medium]** **[Facebook, Amazon, Microsoft]**

Practice link: <https://leetcode.com/problems/find-all-duplicates-in-an-array/>

2. Implement next permutation, which rearranges numbers into the lexicographically next greater permutation of numbers.

If such an arrangement is not possible, it must rearrange it as the lowest possible order (i.e., sorted in ascending order).

The replacement must be in place and use only constant extra memory.

**[Medium] [Facebook, Amazon, Rubrik, Microsoft]**

Practice link: <https://leetcode.com/problems/next-permutation/>

3. Given an array of integers nums sorted in ascending order, find the starting and ending position of a given target value.

If target is not found in the array, return [-1, -1].

You must write an algorithm with  $O(\log n)$  runtime complexity.

**[Medium] [Facebook, Amazon, LinkedIn, Uber]**

Practice link: <https://leetcode.com/problems/find-first-and-last-position-of-element-in-sorted-array/>