

Easy

1) Build Array from Permutation (Leetcode 1920)

Given a **zero-based permutation** `nums` (**0-indexed**), build an array `ans` of the **same length** where `ans[i] = nums[nums[i]]` for each $0 \leq i < \text{nums.length}$ and return it.

A **zero-based permutation** `nums` is an array of **distinct** integers from 0 to `nums.length - 1` (**inclusive**).

Example 1:

Input: `nums = [0,2,1,5,3,4]`

Output: `ans: [0,1,2,4,5,3]`

Explanation: The array `ans` is built as follows:

```
ans = [nums[nums[0]], nums[nums[1]], nums[nums[2]], nums[nums[3]], nums[nums[4]], nums[nums[5]]]
     = [nums[0], nums[2], nums[1], nums[5], nums[3], nums[4]]
     = [0,1,2,4,5,3]
```

2) Running Sum of 1D array(Leetcode 1480)

Given an array `nums`. We define a running sum of an array as `runningSum[i] = sum(nums[0]...nums[i])`.

Return the running sum of `nums`.

Example 1:

Input: `nums = [1,2,3,4]`

Output: `[1,3,6,10]`

Explanation: Running sum is obtained as follows: `[1, 1+2, 1+2+3, 1+2+3+4]`.

3) Shuffle the array (Leetcode 1470)

Given the array `nums` consisting of $2n$ elements in the form $[x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]$.

Return the array in the form $[x_1, y_1, x_2, y_2, \dots, x_n, y_n]$.

Example 1:

Input: `nums = [2,5,1,3,4,7]`, `n = 3`

Output: `[2,3,5,4,1,7]`

Explanation: Since $x_1=2$, $x_2=5$, $x_3=1$, $y_1=3$, $y_2=4$, $y_3=7$ then the answer is `[2,3,5,4,1,7]`.

4) Number of Good Pairs(Leetcode 1512)

Given an array of integers `nums`, return *the number of good pairs*.

A pair (i, j) is called *good* if `nums[i] == nums[j]` and $i < j$.

Example 1:

Input: `nums = [1,2,3,1,1,3]`

Output: 4

Explanation: There are 4 good pairs $(0,3)$, $(0,4)$, $(3,4)$, $(2,5)$

5) Squares of a Sorted Array(Leetcode 977)

Given an integer array `nums` sorted in non-decreasing order, return *an array of the squares of each number sorted in non-decreasing order*.

Example 1:

Input: `nums = [-4,-1,0,3,10]`

Output: `[0,1,9,16,100]`

Explanation: After squaring, the array becomes `[16,1,0,9,100]`.

After sorting, it becomes `[0,1,9,16,100]`.

6)Count Pairs Whose Sum is Less than Target(Leetcode 2824)

Given a 0-indexed integer array `nums` of length `n` and an integer `target`, return *the number of pairs (i, j) where $0 \leq i < j < n$ and $nums[i] + nums[j] < target$* .

Example 1:

Input: `nums = [-1,1,2,3,1]`, `target = 2`

Output: 3

Explanation: There are 3 pairs of indices that satisfy the conditions in the statement:

- (0, 1) since $0 < 1$ and $nums[0] + nums[1] = 0 < target$
- (0, 2) since $0 < 2$ and $nums[0] + nums[2] = 1 < target$
- (0, 4) since $0 < 4$ and $nums[0] + nums[4] = 0 < target$

Note that (0, 3) is not counted since $nums[0] + nums[3]$ is not strictly less than the target.

7)Single Element(Leetcode 136)

Given a **non-empty** array of integers nums, every element appears *twice* except for one. Find that single one.

You must implement a solution with a linear runtime complexity and use only constant extra space.

Example 1:

Input: nums = [2,2,1]

Output: 1

8)Find Numbers with Even Number of Digit(Leetcode 1295)

Given an array nums of integers, return how many of them contain an even number of digits.

Example 1:

Input: nums = [12,345,2,6,7896]

Output: 2

Explanation:

12 contains 2 digits (even number of digits).

345 contains 3 digits (odd number of digits).

2 contains 1 digit (odd number of digits).

6 contains 1 digit (odd number of digits).

7896 contains 4 digits (even number of digits).

Therefore only 12 and 7896 contain an even number of digits.

Medium

9) Find First and Last Position of Element in Sorted Array (Leetcode 34)

Given an array of integers `nums` sorted in non-decreasing order, find the starting and ending position of a given target value.

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

Example 1:

Input: `nums = [5,7,7,8,8,10]`, `target = 8`

Output: `[3,4]`

10) Maximum Subarray (Leetcode 53)

Given an integer array `nums`, find the subarray (A subarray is a contiguous non-empty sequence of elements within an array.) with the largest sum, and return *its sum*.

Example 1:

Input: `nums = [-2,1,-3,4,-1,2,1,-5,4]`

Output: 6

Explanation: The subarray `[4,-1,2,1]` has the largest sum 6.

