Problem 3731: Find Missing Elements

Problem Information

Difficulty: Easy
Acceptance Rate: 81.57%
Paid Only: No
Tags: Array, Hash Table, Sorting

Problem Description

You are given an integer array `nums` consisting of **unique** integers.

Originally, `nums` contained **every integer** within a certain range. However, some integers might have gone **missing** from the array.

The **smallest** and **largest** integers of the original range are still present in `nums`.

Return a **sorted** list of all the missing integers in this range. If no integers are missing, return an **empty** list.

Example 1:

Input: nums = [1,4,2,5]

Output: [3]

Explanation:

The smallest integer is 1 and the largest is 5, so the full range should be `[1,2,3,4,5]`. Among these, only 3 is missing.

Example 2:

Input: nums = [7,8,6,9]

Output: []

Explanation:

The smallest integer is 6 and the largest is 9, so the full range is `[6,7,8,9]`. All integers are already present, so no integer is missing.

Example 3:

Input: nums = [5,1]

Output: [2,3,4]

Explanation:

The smallest integer is 1 and the largest is 5, so the full range should be `[1,2,3,4,5]`. The missing integers are 2, 3, and 4.

Constraints:

* `2 <= nums.length <= 100` * `1 <= nums[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
vector<int> findMissingElements(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public List<Integer> findMissingElements(int[] nums) {

}
```

```
}
```

Python3:

```
class Solution:
def findMissingElements(self, nums: List[int]) -> List[int]:
```