# Problem 2150: Find All Lonely Numbers in the Array

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 62.28%
**Paid Only:** No
**Tags:** Array, Hash Table, Counting

## Problem Description

You are given an integer array `nums`. A number `x` is **lonely** when it appears only **once** , and no **adjacent** numbers (i.e. `x + 1` and `x - 1)` appear in the array.

Return _**all** lonely numbers in _`nums`. You may return the answer in **any order**.

**Example 1:**

**Input:** nums = [10,6,5,8] **Output:** [10,8] **Explanation:** - 10 is a lonely number since it appears exactly once and 9 and 11 does not appear in nums. - 8 is a lonely number since it appears exactly once and 7 and 9 does not appear in nums. - 5 is not a lonely number since 6 appears in nums and vice versa. Hence, the lonely numbers in nums are [10, 8]. Note that [8, 10] may also be returned.

**Example 2:**

**Input:** nums = [1,3,5,3] **Output:** [1,5] **Explanation:** - 1 is a lonely number since it appears exactly once and 0 and 2 does not appear in nums. - 5 is a lonely number since it appears exactly once and 4 and 6 does not appear in nums. - 3 is not a lonely number since it appears twice. Hence, the lonely numbers in nums are [1, 5]. Note that [5, 1] may also be returned.

**Constraints:**

* `1 <= nums.length <= 105` * `0 <= nums[i] <= 106`

## Code Snippets

**C++:**

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

}
};
```

**Java:**

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

}
}
```

**Python3:**

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