5990. Find All Lonely Numbers in the Array

My Submissions (/contest/weekly-contest-277/problems/find-all-lonely-numbers-in-the-array/submissions/)

Back to Contest (/contest/weekly-contest-277/)

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**.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

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
- 8 is a lonely number since it appears exactly once and 7 and 9 does no
- 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 <= 10^5`
- `0 <= nums[i] <= 10^6`

JavaScript

```
const counter = (a_or_s) => { let m = new Map(); for (const x of a_or_s) m.set(x, m.get(x) + 1 || 1); return m; };

const findLonely = (a) => {
    let m = counter(a), res = [];
    for (const x of a) {
        if (m.get(x) == 1 && !m.has(x - 1) && !m.has(x + 1)) res.push(x);
    }
    return res;
};
```

Custom Testcase Use Example Testcases

Run Submit

Submission Result: Accepted (/submissions/detail/625707399/)

More Details (/submissions/detail/625707399/)

Share your acceptance!

1