6258. Longest Square Streak in an Array

My Submissions (/contest/weekly-contest-323/problems/longest-square-streak-in-an-array/submissions/) Back to Contest (/contest/weekly-contest-323/)

You are given an integer array `nums`. A subsequence of `nums` is called a **square streak** if:

- The length of the subsequence is at least `2`, and
- **after** sorting the subsequence, each element (except the first element) is the **square** of the previous number.

Return *the length of the **longest square streak** in* `nums`, *or return* `-1` *if there is no **square streak**.*

A **subsequence** is an array that can be derived from another array by deleting some or no elements without changing the order of the remaining elements.

User Accepted:	1
User Tried:	2
Total Accepted:	1
Total Submissions:	2
Difficulty:	Medium

Example 1:

```
Input: nums = [4,3,6,16,8,2]
Output: 3
Explanation: Choose the subsequence [4,16,2]. After sorting it, it becomes [2,4,16].
- 4 = 2 * 2.
- 16 = 4 * 4.
Therefore, [4,16,2] is a square streak.
It can be shown that every subsequence of length 4 is not a square streak.
```

Example 2:

```
Input: nums = [2,3,5,6,7]
Output: -1
Explanation: There is no square streak in nums so return -1.
```

Constraints:

- `2 <= nums.length <= 10^5`
- `2 <= nums[i] <= 10^5`

JavaScript

```
const counter_value_in_indexA_in = (a_or_s) => { let m = new Map(); let n = a_or_s.length; for (let i = 0; i < n; i++) {
if (!m.has(a_or_s[i])) m.set(a_or_s[i], []); m.get(a_or_s[i]).push(i); } return m; };
const stmkey_in = (m) => new Map([...m].sort((x, y) => x[0] - y[0]));

function Bisect() {
    return { insort_right, insort_left, bisect_left, bisect_right }
    function insort_right(a, x, lo = 0, hi = null) {
        lo = bisect_right(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_right(a, x, lo = 0, hi = null) { // > upper_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] > x ? hi = mid : lo = mid + 1;
        }
        return lo;
    }
    function insort_left(a, x, lo = 0, hi = null) {
        lo = bisect_left(a, x, lo, hi);
        a.splice(lo, 0, x);
    }
    function bisect_left(a, x, lo = 0, hi = null) { // >= lower_bound
        if (lo < 0) throw new Error('lo must be non-negative');
        if (hi == null) hi = a.length;
        while (lo < hi) {
            let mid = parseInt((lo + hi) / 2);
            a[mid] < x ? lo = mid + 1 : hi = mid;
        }
```

```
        return lo;
    }
}

const longestSquareStreak = (a) => {
    a.sort((x, y) => x - y);
    let m = counter_value_in_indexA_in(a), bi = new Bisect(), res = -1;
    m = stmkey_in(m);
    // pr(m)
    for (const [x, a] of m) {
        let cur = x, d = [x], pre = a[0];
        while (1) {
            let next = cur ** 2;
            // pr("next", next);
            if (m.has(next)) {
                let b = m.get(next), idx = bi.bisect_right(b, pre);
                // pr("b", b, idx)
                if (idx == b.length) {
                    break;
                } else {
                    d.push(next);
                    pre = b[idx];
                    cur = next;
                }
            } else {
                break;
            }
        }
        // pr(x, d);
        if (d.length >= 2) res = Math.max(res, d.length);
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/857898949/) | More Details > (/submissions/detail/857898949/)

Share your acceptance!

1