Longest Square Streak In An Array

solved by	Senan
Platform	LeetCode
difficulty	Medium
# Serial	2501
tags	Vector set
language	C++
solved on	@28/10/2024
link	https://leetcode.com/problems/longest-square-streak-in-an-array/description/
Completion	☑

Intuition

The problem involves finding the longest "square streak" for each element in an array, where a "square streak" is defined by consecutive elements that are perfect squares of the previous element. Using a set to store elements allows for constant-time lookup, making it efficient to check if each squared value exists in the array.

Approach

1. Convert the array into an unordered set for O(1) lookup time.
2. Iterate over each element in the array. For each element:
    - Start a counter to keep track of the current streak length.
    - Keep squaring the element until it’s no longer in the set or exceeds a certain threshold (in this case, `100,000`).
3. Update the maximum streak if the current streak exceeds the longest recorded streak.
4. Return the longest streak if it’s more than 1; otherwise, return `1`.

Complexity

Time Complexity:

- **O(N)**, where **N** is the number of elements in the input vector. In the worst case, each element could be squared up to a threshold.

Space Complexity:

- **O(N)** for storing elements in the set.

Code

```
class Solution {
public:
    int longestSquareStreak(vector<int>& nums) {
        unordered_set<int> mySet(nums.begin(), nums.end());
        int maxStreak = 0;
```

```
        for(auto elem: nums){
            int currStreak = 0;
            long long currVal = elem;
            while(mySet.find(currVal) != mySet.end()){
                currVal *= currVal;
                maxStreak = max(maxStreak, ++currStreak);
                if(currVal > 100000) break;
            }
        }
        return (maxStreak > 1) ? maxStreak : -1;
    }
};
```