Problem 2501: Longest Square Streak in an Array

Problem Information

Difficulty: Medium
Acceptance Rate: 53.08%
Paid Only: No
Tags: Array, Hash Table, Binary Search, Dynamic Programming, Sorting

Problem Description

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.

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 <= 105` * `2 <= nums[i] <= 105`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public int longestSquareStreak(int[] nums) {

}
}
```

Python3:

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