Problem 2778: Sum of Squares of Special Elements

Problem Information

Difficulty: Easy
Acceptance Rate: 81.62%
Paid Only: No
Tags: Array, Enumeration

Problem Description

You are given a **1-indexed** integer array `nums` of length `n`.

An element `nums[i]` of `nums` is called **special** if `i` divides `n`, i.e. `n % i == 0`.

Return _the**sum of the squares** of all **special** elements of _`nums`.

Example 1:

Input: nums = [1,2,3,4] **Output:** 21 **Explanation:** There are exactly 3 special elements in nums: nums[1] since 1 divides 4, nums[2] since 2 divides 4, and nums[4] since 4 divides 4. Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[4] * nums[4] = 1 * 1 + 2 * 2 + 4 * 4 = 21.

Example 2:

Input: nums = [2,7,1,19,18,3] **Output:** 63 **Explanation:** There are exactly 4 special elements in nums: nums[1] since 1 divides 6, nums[2] since 2 divides 6, nums[3] since 3 divides 6, and nums[6] since 6 divides 6. Hence, the sum of the squares of all special elements of nums is nums[1] * nums[1] + nums[2] * nums[2] + nums[3] * nums[3] + nums[6] * nums[6] = 2 * 2 + 7 * 7 + 1 * 1 + 3 * 3 = 63.

Constraints:

* `1 <= nums.length == n <= 50` * `1 <= nums[i] <= 50`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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