Problem 3411: Maximum Subarray With Equal Products

Problem Information

Difficulty: Easy
Acceptance Rate: 46.07%
Paid Only: No
Tags: Array, Math, Sliding Window, Enumeration, Number Theory

Problem Description

You are given an array of **positive** integers `nums`.

An array `arr` is called **product equivalent** if `prod(arr) == lcm(arr) * gcd(arr)`, where:

* `prod(arr)` is the product of all elements of `arr`. * `gcd(arr)` is the GCD of all elements of `arr`. * `lcm(arr)` is the LCM of all elements of `arr`.

Return the length of the **longest** **product equivalent** subarray of `nums`.

Example 1:

Input: nums = [1,2,1,2,1,1,1]

Output: 5

Explanation:

The longest product equivalent subarray is `[1, 2, 1, 1, 1]`, where `prod([1, 2, 1, 1, 1]) = 2`, `gcd([1, 2, 1, 1, 1]) = 1`, and `lcm([1, 2, 1, 1, 1]) = 2`.

Example 2:

Input: nums = [2,3,4,5,6]

Output: 3

Explanation:

The longest product equivalent subarray is `[3, 4, 5].`

Example 3:

Input: nums = [1,2,3,1,4,5,1]

Output: 5

Constraints:

* `2 <= nums.length <= 100` * `1 <= nums[i] <= 10`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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