Problem 2436: Minimum Split Into Subarrays With GCD Greater Than One

Problem Information

Difficulty: Medium
Acceptance Rate: 70.24%
Paid Only: Yes
Tags: Array, Math, Dynamic Programming, Greedy, Number Theory

Problem Description

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

Split the array into **one or more** disjoint subarrays such that:

* Each element of the array belongs to **exactly one** subarray, and * The **GCD** of the elements of each subarray is strictly greater than `1`.

Return _the minimum number of subarrays that can be obtained after the split_.

Note that:

* The **GCD** of a subarray is the largest positive integer that evenly divides all the elements of the subarray. * A **subarray** is a contiguous part of the array.

Example 1:

Input: nums = [12,6,3,14,8] **Output:** 2 **Explanation:** We can split the array into the subarrays: [12,6,3] and [14,8]. - The GCD of 12, 6 and 3 is 3, which is strictly greater than 1. - The GCD of 14 and 8 is 2, which is strictly greater than 1. It can be shown that splitting the array into one subarray will make the GCD = 1.

Example 2:

Input: nums = [4,12,6,14] **Output:** 1 **Explanation:** We can split the array into only one subarray, which is the whole array.

Constraints:

* `1 <= nums.length <= 2000` * `2 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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