Problem 3326: Minimum Division Operations to Make Array Non Decreasing

Problem Information

Difficulty: Medium
Acceptance Rate: 28.91%
Paid Only: No
Tags: Array, Math, Greedy, Number Theory

Problem Description

You are given an integer array `nums`.

Any **positive** divisor of a natural number `x` that is **strictly less** than `x` is called a **proper divisor** of `x`. For example, 2 is a _proper divisor_ of 4, while 6 is not a _proper divisor_ of 6.

You are allowed to perform an **operation** any number of times on `nums`, where in each **operation** you select any _one_ element from `nums` and divide it by its **greatest** **proper divisor**.

Return the **minimum** number of **operations** required to make the array **non-decreasing**.

If it is **not** possible to make the array _non-decreasing_ using any number of operations, return `-1`.

Example 1:

Input: nums = [25,7]

Output: 1

Explanation:

Using a single operation, 25 gets divided by 5 and `nums` becomes `[5, 7]`.

Example 2:

Input: nums = [7,7,6]

Output: -1

Example 3:

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

Output: 0

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 106`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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