Problem 3717: Minimum Operations to Make the Array Beautiful

Problem Information

Difficulty: Medium
Acceptance Rate: 41.10%
Paid Only: Yes
Tags: Array, Dynamic Programming

Problem Description

You are given an integer array `nums`.

An array is called **beautiful** if for every index `i > 0`, the value at `nums[i]` is **divisible** by `nums[i - 1]`.

In one operation, you may **increment** any element `nums[i]` (with `i > 0`) by `1`.

Return the **minimum number of operations** required to make the array beautiful.

Example 1:

Input: nums = [3,7,9]

Output: 2

Explanation:

Applying the operation twice on `nums[1]` makes the array beautiful: `[3,9,9]`

Example 2:

Input: nums = [1,1,1]

Output: 0

Explanation:

The given array is already beautiful.

Example 3:

Input: nums = [4]

Output: 0

Explanation:

The array has only one element, so it's already beautiful.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i] <= 50■■■`

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:
```