Problem 2366: Minimum Replacements to Sort the Array

Problem Information

Difficulty: Hard
Acceptance Rate: 53.18%
Paid Only: No
Tags: Array, Math, Greedy

Problem Description

You are given a **0-indexed** integer array `nums`. In one operation you can replace any element of the array with **any two** elements that **sum** to it.

* For example, consider `nums = [5,6,7]`. In one operation, we can replace `nums[1]` with `2` and `4` and convert `nums` to `[5,2,4,7]`.

Return _the minimum number of operations to make an array that is sorted in**non-decreasing** order_.

Example 1:

Input: nums = [3,9,3] **Output:** 2 **Explanation:** Here are the steps to sort the array in non-decreasing order: - From [3,9,3], replace the 9 with 3 and 6 so the array becomes [3,3,6,3] - From [3,3,6,3], replace the 6 with 3 and 3 so the array becomes [3,3,3,3,3] There are 2 steps to sort the array in non-decreasing order. Therefore, we return 2.

Example 2:

Input: nums = [1,2,3,4,5] **Output:** 0 **Explanation:** The array is already in non-decreasing order. Therefore, we return 0.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109`

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public long minimumReplacement(int[] nums) {

}
}
```

Python3:

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