Problem 2449: Minimum Number of Operations to Make Arrays Similar

Problem Information

Difficulty: Hard
Acceptance Rate: 61.23%
Paid Only: No
Tags: Array, Greedy, Sorting

Problem Description

You are given two positive integer arrays `nums` and `target`, of the same length.

In one operation, you can choose any two **distinct** indices `i` and `j` where `0 <= i, j < nums.length` and:

* set `nums[i] = nums[i] + 2` and * set `nums[j] = nums[j] - 2`.

Two arrays are considered to be **similar** if the frequency of each element is the same.

Return _the minimum number of operations required to make_`nums` _similar to_`target`. The test cases are generated such that `nums` can always be similar to `target`.

Example 1:

Input: nums = [8,12,6], target = [2,14,10] **Output:** 2 **Explanation:** It is possible to make nums similar to target in two operations: - Choose i = 0 and j = 2, nums = [10,12,4]. - Choose i = 1 and j = 2, nums = [10,14,2]. It can be shown that 2 is the minimum number of operations needed.

Example 2:

Input: nums = [1,2,5], target = [4,1,3] **Output:** 1 **Explanation:** We can make nums similar to target in one operation: - Choose i = 1 and j = 2, nums = [1,4,3].

Example 3:

Input: nums = [1,1,1,1,1], target = [1,1,1,1,1] **Output:** 0 **Explanation:** The array nums is already similiar to target.

Constraints:

* `n == nums.length == target.length` * `1 <= n <= 105` * `1 <= nums[i], target[i] <= 106` * It is possible to make `nums` similar to `target`.

Code Snippets

C++:

```
class Solution {
public:
long long makeSimilar(vector<int>& nums, vector<int>& target) {

}
};
```

Java:

```
class Solution {
public long makeSimilar(int[] nums, int[] target) {

}
}
```

Python3:

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