(/) Day 27 Explore Problems(/problemset/all/) Mock(/interview/) (/contest/) Contest Discuss(/discuss/) Limited time event to win giveaway! (/track/?Storedata=eyJ1cmwiOiAiaHR0cHM6Ly9sZWV0Y29kZS5jb20vZGlzY3Vzcy9nZW5lcmFsLWRR

5691. Equal Sum Arrays With Minimum Number of Operations

My Submissions (/contest/weekly-contest-230/problems/equal-sum-arrays-with-minimum-number-of-operations/submissions/)

Back to Contest (/contest/weekly-contest-230/)

You are given two arrays of integers `nums1` and `nums2`, possibly of different lengths. The values in the arrays are between `1` and `6`, inclusive.

In one operation, you can change any integer's value in **any** of the arrays to **any** value between `1` and `6`, inclusive.

Return *the minimum number of operations required to make the sum of values in* `nums1` *equal to the sum of values in* `nums2`. Return `-1` if it is not possible to make the sum of the two arrays equal.

User Accepted:	142
User Tried:	178
Total Accepted:	142
Total Submissions:	218
Difficulty:	Medium

Example 1:

```
Input: nums1 = [1,2,3,4,5,6], nums2 = [1,1,2,2,2,2]
Output: 3
Explanation: You can make the sums of nums1 and nums2 equal with 3 operations. All indices are 0-indexed.
- Change nums2[0] to 6. nums1 = [1,2,3,4,5,6], nums2 = [6,1,2,2,2,2].
- Change nums1[5] to 1. nums1 = [1,2,3,4,5,1], nums2 = [6,1,2,2,2,2].
- Change nums1[2] to 2. nums1 = [1,2,2,4,5,1], nums2 = [6,1,2,2,2,2].
```

Example 2:

```
Input: nums1 = [1,1,1,1,1,1,1], nums2 = [6]
Output: -1
Explanation: There is no way to decrease the sum of nums1 or to increase the sum of nums2 to make them equal.
```

Example 3:

```
Input: nums1 = [6,6], nums2 = [1]
Output: 3
Explanation: You can make the sums of nums1 and nums2 equal with 3 operations. All indices are 0-indexed.
- Change nums1[0] to 2. nums1 = [2,6], nums2 = [1].
- Change nums1[1] to 2. nums1 = [2,2], nums2 = [1].
- Change nums2[0] to 4. nums1 = [2,2], nums2 = [4].
```

Constraints:

- `1 <= nums1.length, nums2.length <=` 10^5
- `1 <= nums1[i], nums2[i] <= 6`

JavaScript

```
/**
 * @param {number[]} nums1
 * @param {number[]} nums2
 * @return {number}
 */
var minOperations = function(nums1, nums2) {

};
```