Minimum Operations to Make Array Equal II

You are given two integer arrays `nums1` and `nums2` of equal length `n` and an integer `k`. You can perform the following operation on `nums1`:

- Choose two indexes `i` and `j` and increment `nums1[i]` by `k` and decrement `nums1[j]` by `k`. In other words, `nums1[i] = nums1[i] + k` and `nums1[j] = nums1[j] - k`.

`nums1` is said to be **equal** to `nums2` if for all indices `i` such that `0 <= i < n`, `nums1[i] == nums2[i]`.

Return *the **minimum** number of operations required to make* `nums1` *equal to* `nums2`. If it is impossible to make them equal, return `-1`.

Example 1:

```
Input: nums1 = [4,3,1,4], nums2 = [1,3,7,1], k = 3

Output: 2

Explanation: In 2 operations, we can transform nums1 to nums2.

1st operation: i = 2, j = 0. After applying the operation, nums1 = [1,3,4,4].

2nd operation: i = 2, j = 3. After applying the operation, nums1 = [1,3,7,1].

One can prove that it is impossible to make arrays equal in fewer operations.
```

Example 2:

```
Input: nums1 = [3,8,5,2], nums2 = [2,4,1,6], k = 1

Output: -1

Explanation: It can be proved that it is impossible to make the two arrays equal.
```

Constraints:

- `n == nums1.length == nums2.length`
- `2 <= n <= ` 10^5
- `0 <= nums1[i], nums2[j] <= ` 10^9
- `0 <= k <= ` 10^5