2499. Minimum Total Cost to Make Arrays Unequal

My Submissions (/contest/biweekly-contest-93/problems/minimum-total-cost-to-make-arrays-unequal/submissions/)

Back to Contest (/contest/biweekly-contest-93/)

You are given two **0-indexed** integer arrays `nums1` and `nums2`, of equal length `n`.

In one operation, you can swap the values of any two indices of `nums1`. The **cost** of this operation is the **sum** of the indices.

Find the **minimum** total cost of performing the given operation **any** number of times such that `nums1[i] != nums2[i]` for all `0 <= i <= n - 1` after performing all the operations.

Return *the **minimum total cost** such that* `nums1` and `nums2` *satisfy the above condition*. In case it is not possible, return `-1`.

User Accepted:	260
User Tried:	726
Total Accepted:	288
Total Submissions:	1410
Difficulty:	Hard

Example 1:

```
Input: nums1 = [1,2,3,4,5], nums2 = [1,2,3,4,5]
Output: 10
Explanation:
One of the ways we can perform the operations is:
- Swap values at indices 0 and 3, incurring cost = 0 + 3 = 3. Now, nums1 = [4,2,3,1,5]
- Swap values at indices 1 and 2, incurring cost = 1 + 2 = 3. Now, nums1 = [4,3,2,1,5].
- Swap values at indices 0 and 4, incurring cost = 0 + 4 = 4. Now, nums1 =[5,3,2,1,4].
We can see that for each index i, nums1[i] != nums2[i]. The cost required here is 10.
Note that there are other ways to swap values, but it can be proven that it is not possible to obtain a cost less than 10.
```

Example 2:

```
Input: nums1 = [2,2,2,1,3], nums2 = [1,2,2,3,3]
Output: 10
Explanation:
One of the ways we can perform the operations is:
- Swap values at indices 2 and 3, incurring cost = 2 + 3 = 5. Now, nums1 = [2,2,1,2,3].
- Swap values at indices 1 and 4, incurring cost = 1 + 4 = 5. Now, nums1 = [2,3,1,2,2].
The total cost needed here is 10, which is the minimum possible.
```

Example 3:

```
Input: nums1 = [1,2,2], nums2 = [1,2,2]
Output: -1
Explanation:
It can be shown that it is not possible to satisfy the given conditions irrespective of the number of operations we perform.
Hence, we return -1.
```

Constraints:

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

Discuss (https://leetcode.com/problems/minimum-total-cost-to-make-arrays-unequal/discuss)

JavaScript

```
const minimumTotalCost = (a, b) => {
    let n = a.length, f = Array(n + 1).fill(0);
    for (let i = 0; i < n; i++) {
        f[a[i]]++;
        f[b[i]]++;
    }
    if (f.some(occ => occ >= n + 1)) return -1;
    let todo = Array(n).fill(false), todoCal = Array(n).fill(0), conflict = 0, res = 0, idx = -1;
```

```
    for (let i = 0; i < n; i++) {
        if (a[i] == b[i]) {
            todo[i] = true;
            todoCal[a[i]]++;
            conflict++;
            res += i;
        }
    }
    for (let i = 1; i <= n; i++) {
        if (todoCal[i] * 2 > conflict) {
            idx = i;
            break;
        }
    }
    if (idx == -1) return res;
    for (let i = 0; i < n; i++) {
        if (!todo[i]) {
            if (todoCal[idx] * 2 > conflict) {
                if (a[i] != idx && b[i] != idx) {
                    conflict++;
                    res += i;
                }
            }
        }
    }
    return res;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/857792565/) | More Details > (/submissions/detail/857792565/)

Share your acceptance!