Problem 3682: Minimum Index Sum of Common Elements

Problem Information

Difficulty: Medium
Acceptance Rate: 79.25%
Paid Only: Yes
Tags: Array, Hash Table

Problem Description

You are given two integer arrays `nums1` and `nums2` of equal length `n`.

We define a pair of indices `(i, j)` as a **good pair** if `nums1[i] == nums2[j]`.

Return the **minimum index sum** `i + j` among all possible good pairs. If no such pairs exist, return -1.

Example 1:

Input: nums1 = [3,2,1], nums2 = [1,3,1]

Output: 1

Explanation:

* Common elements between `nums1` and `nums2` are 1 and 3. * For 3, `[i, j] = [0, 1]`, giving an index sum of `i + j = 1`. * For 1, `[i, j] = [2, 0]`, giving an index sum of `i + j = 2`. * The minimum index sum is 1.

Example 2:

Input: nums1 = [5,1,2], nums2 = [2,1,3]

Output: 2

Explanation:

* Common elements between `nums1` and `nums2` are 1 and 2. * For 1, `[i, j] = [1, 1]`, giving an index sum of `i + j = 2`. * For 2, `[i, j] = [2, 0]`, giving an index sum of `i + j = 2`. * The minimum index sum is 2.

Example 3:

Input: nums1 = [6,4], nums2 = [7,8]

Output: -1

Explanation:

* Since no common elements between `nums1` and `nums2`, the output is -1.

Constraints:

* `1 <= nums1.length == nums2.length <= 105` * `-105 <= nums1[i], nums2[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int minimumSum(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

```
class Solution {
public int minimumSum(int[] nums1, int[] nums2) {

}
}
```

Python3:

```
class Solution:
def minimumSum(self, nums1: List[int], nums2: List[int]) -> int:
```