Problem 760: Find Anagram Mappings

Problem Information

Difficulty: Easy
Acceptance Rate: 83.96%
Paid Only: Yes
Tags: Array, Hash Table

Problem Description

You are given two integer arrays `nums1` and `nums2` where `nums2` is **an anagram** of `nums1`. Both arrays may contain duplicates.

Return _an index mapping array_`mapping` _from_`nums1` _to_`nums2` _where_`mapping[i] = j` _means the_`ith` _element in_`nums1` _appears in_`nums2` _at index_`j`. If there are multiple answers, return **any of them**.

An array `a` is **an anagram** of an array `b` means `b` is made by randomizing the order of the elements in `a`.

Example 1:

Input: nums1 = [12,28,46,32,50], nums2 = [50,12,32,46,28] **Output:** [1,4,3,2,0] **Explanation:** As mapping[0] = 1 because the 0th element of nums1 appears at nums2[1], and mapping[1] = 4 because the 1st element of nums1 appears at nums2[4], and so on.

Example 2:

Input: nums1 = [84,46], nums2 = [84,46] **Output:** [0,1]

Constraints:

* `1 <= nums1.length <= 100` * `nums2.length == nums1.length` * `0 <= nums1[i], nums2[i] <= 105` * `nums2` is an anagram of `nums1`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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