Problem 3690: Split and Merge Array Transformation

Problem Information

Difficulty: Medium
Acceptance Rate: 52.15%
Paid Only: No
Tags: Array, Hash Table, Breadth-First Search

Problem Description

You are given two integer arrays `nums1` and `nums2`, each of length `n`. You may perform the following **split-and-merge operation** on `nums1` any number of times:

1. Choose a subarray `nums1[L..R]`. 2. Remove that subarray, leaving the prefix `nums1[0..L-1]` (empty if `L = 0`) and the suffix `nums1[R+1..n-1]` (empty if `R = n - 1`). 3. Re-insert the removed subarray (in its original order) at **any** position in the remaining array (i.e., between any two elements, at the very start, or at the very end).

Return the **minimum** number of **split-and-merge operations** needed to transform `nums1` into `nums2`.

Example 1:

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

Output: 1

Explanation:

* Split out the subarray `[3]` (`L = 0`, `R = 0`); the remaining array is `[1,2]`. * Insert `[3]` at the end; the array becomes `[1,2,3]`.

Example 2:

Input: nums1 = [1,1,2,3,4,5], nums2 = [5,4,3,2,1,1]

Output: 3

Explanation:

* Remove `[1,1,2]` at indices `0 - 2`; remaining is `[3,4,5]`; insert `[1,1,2]` at position `2`, resulting in `[3,4,1,1,2,5]`. * Remove `[4,1,1]` at indices `1 - 3`; remaining is `[3,2,5]`; insert `[4,1,1]` at position `3`, resulting in `[3,2,5,4,1,1]`. * Remove `[3,2]` at indices `0 - 1`; remaining is `[5,4,1,1]`; insert `[3,2]` at position `2`, resulting in `[5,4,3,2,1,1]`.

Constraints:

* `2 <= n == nums1.length == nums2.length <= 6` * `-105 <= nums1[i], nums2[i] <= 105` * `nums2` is a **permutation** of `nums1`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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