Problem 718: Maximum Length of Repeated Subarray

Problem Information

Difficulty: Medium
Acceptance Rate: 51.16%
Paid Only: No
Tags: Array, Binary Search, Dynamic Programming, Sliding Window, Rolling Hash, Hash Function

Problem Description

Given two integer arrays `nums1` and `nums2`, return _the maximum length of a subarray that appears in**both** arrays_.

Example 1:

Input: nums1 = [1,2,3,2,1], nums2 = [3,2,1,4,7] **Output:** 3 **Explanation:** The repeated subarray with maximum length is [3,2,1].

Example 2:

Input: nums1 = [0,0,0,0,0], nums2 = [0,0,0,0,0] **Output:** 5 **Explanation:** The repeated subarray with maximum length is [0,0,0,0,0].

Constraints:

* `1 <= nums1.length, nums2.length <= 1000` * `0 <= nums1[i], nums2[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
```

```
int findLength(vector<int>& nums1, vector<int>& nums2) {

}
};
```

Java:

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

}
}
```

Python3:

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