718. Maximum Length of Repeated Subarray

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
```

Constraints:

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

```
class Solution:
    def findLength(self, nums1: List[int], nums2: List[int]) -> int:
        n = len(nums1)
        m = len(nums2)
        dp = [[0]*(m+1) for i in range(n+1)]
        maxL = 0
        for i in range(1,n+1):
            for j in range(1,m+1):
                if nums1[i-1]==nums2[j-1]:
                    dp[i][j] = 1+dp[i-1][j-1]
                    maxL = max(maxL,dp[i][j])

        return maxL
```