Rotate Array LeetCode

You are given an array of integers **nums** and an integer **k**. You need to rotate the array to the right by **k** steps.

Input:

nums = [1, 1, 3, 4, 9, 10, 12]

k = 4

Output:

The array before Rotating: 1 1 3 4 9 10 12

The array after Rotating from index 4: 4 9 10 12 1 1 3

Approach 1: Function to rotate the array using temporary array

- The **rotate** function uses a temporary array **temp** to store the rotated elements.
- It iterates through the input array and assigns the elements to the corresponding rotated positions in **temp**.
- Finally, it copies the rotated elements back to the original array.
- **Time Complexity: O(n), where n is the size of the array. We iterate through the array once.**
- **Space Complexity: O(n), as we use an additional temporary array.**

Approach 2: Optimized function to rotate the array in-place

- The **rotateOptimized** function performs the rotation in-place without using an additional temporary array.
- It reverses the entire array, then reverses the first **k** elements, and finally reverses the remaining elements.
- The three reversal steps achieve the rotation effect.
- **Time Complexity: O(n), where n is the size of the array. We reverse the array and perform element swaps in-place.**
- **Space Complexity: O(1), as we don't use any additional space except for the input array.**

Which Approach to Use:

- Approach 2 (In-place Reversal) is generally more efficient and preferred since it doesn't require extra space.
- Approach 1 (Temporary Array) can be useful in scenarios where preserving the original array is essential, or if the input array is read-only.