Problem Link:

https://leetcode.com/problems/rotate-array/

Problem Description:

Given an integer array nums, rotate the array to the right by k steps, where k is non-negative.

Problem Approach:

We can write an algorithm to rotate an array k times, but that won't be optimal, as it may fail if the test case is too big, due to its time complexity being O(n^2). So, we'll find an optimal solution using a three-step reversal algorithm.

Sample Test Case:

Input: nums = [1,2,3,4,5,6,7], k = 3

Output: [5,6,7,1,2,3,4]

Solution:

The solution is quite simple. First, we minimize k by setting it as remainder of when we divide k by length of the array. As 1st rotation would be the same as 6th rotation in an array of size 5. Then, we first reverse the entire list. So, using the above test case, at this point, the temp array would look like:

[7,6,5,4,3,2,1]

Then, we reverse the first k elements of the list, which would give us:

[5,6,7,4,3,2,1]

Then, we reverse the last n-k elements, which would give us:

[5,6,7,1,2,3,4]

Which is our answer! It can also be verified by rotating the array k times. The time complexity of this is O(n), and the space complexity is O(1). However, there's a smaller solution using slices in python, but for that, the space complexity is O(n). Hence, not the best choice.

Code (Python):

```python
def rotate(self, nums: List[int], k: int) -> None:
    n = len(nums)
    k = k % n
    temp = [0] * n
    for i in range(n-1, -1, -1):
        temp[n-i-1] = nums[i]
    for i in range(n - k):
        temp[k + i] = nums[i]
    for i in range(k):
        temp[i] = nums[n - k + i]
    for i in range(n):
        nums[i] = temp[i]
```

In the following solution, we're doing the same thing, but with the help of slices. nums[-k:] gives us the last k elements in reversed manner, and nums[:-k] gives us the first n-k elements (n refers to the size of the array). Adding the two slices gives us the expected answer.

```python
def rotate(self, nums: List[int], k: int) -> None:
    k = k%len(nums)
    nums[:] = nums[-k:] + nums[:-k]
```