189 Rotate Array

2018年4月3日 20:11

Question:

Rotate an array of *n* elements to the right by *k* steps.
For example, with *n* = 7 and *k* = 3, the array [1,2,3,4,5,6,7] is rotated to [5,6,7,1,2,3,4].

Note:

Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.
[show hint]
Related problem: Reverse Words in a String II

来自 <https://leetcode.com/problems/rotate-array/description/>

将包含 n 个元素的数组向右旋转 k 步。
例如，如果 $n = 7$ ， $k = 3$，给定数组 [1,2,3,4,5,6,7] ，向右旋转后的结果为 [5,6,7,1,2,3,4]。

注意:

尽可能找到更多的解决方案，这里最少有三种不同的方法解决这个问题。
[显示提示]

提示:

要求空间复杂度为 O(1)。

Solution for Python3:

```python
class Solution1:
    def rotate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        n = len(nums)
        k %= n
        nums[:] = nums[-k:] + nums[:-k]

class Solution2:
    def rotate(self, nums, k):
        """
        :type nums: List[int]
        :type k: int
        :rtype: void Do not return anything, modify nums in-place instead.
        """
        n, s = len(nums), 0
        k %= n
        while k:
            for i in range(k):
                nums[s + i], nums[n - k + i] = nums[n - k + i], nums[s + i]
            s += k
            k %= (n - s)
```

Solution for C++:

```cpp
class Solution1 {
public:
    void rotate(vector<int>& nums, int k) {
        if (nums.empty() || k % nums.size() == 0) {
            return;
        }
        int n = nums.size();
        k %= n; //k有可能超过数组大小，所以要取余
        //翻转前n-k部分，下标范围[0,n-k-1]
        reverse(nums.begin(), nums.begin() + n - k);
        //翻转后k部分，下标范围[n-k,n-1]
        reverse(nums.begin() + n - k, nums.end());
        reverse(nums.begin(), nums.end());//翻转整个数组，下标范围[0,n-1]
    }
};

class Solution2 {
public:
    void rotate(vector<int>& nums, int k) {
        if (nums.empty() || k % nums.size() == 0) {
            return;
        }
        int n = nums.size();
        reverse(nums.begin(), nums.end());
        reverse(nums.begin(), nums.begin() + k % n);
        reverse(nums.begin() + k % n, nums.end());
    }
};

class Solution3 {
public:
    void rotate(vector<int>& nums, int k) {
        int n = nums.size();
        k %= n;
        for (int s = 0; k = k % (n - s); s += k) {
            for (int i = 0; i < k; i++) {
                swap(nums[s + i], nums[n - k + i]);
            }
        }
    }
};
```