

第38题——旋转数组

给定一个数组，将数组中的元素向右移动 k 个位置，其中 k 是非负数。

进阶：

尽可能想出更多的解决方案，至少有三种不同的方法可以解决这个问题。你可以使用空间复杂度为 $O(1)$ 的 原地 算法解决这个问题吗？

示例 1:

输入：nums = [1,2,3,4,5,6,7], k = 3

输出：[5,6,7,1,2,3,4]

解释:

向右旋转 1 步：[7,1,2,3,4,5,6]
向右旋转 2 步：[6,7,1,2,3,4,5]
向右旋转 3 步：[5,6,7,1,2,3,4]

示例 2:

输入：nums = [-1,-100,3,99], k = 2

输出：[3,99,-1,-100]

解释:

向右旋转 1 步：[99,-1,-100,3]
向右旋转 2 步：[3,99,-1,-100]

提示:

```
1 <= nums.length <= 2 * 104  
-231 <= nums[i] <= 231 - 1  
0 <= k <= 105
```

反转三次数组(1)

1. 假设数组为 1 2 3 4 5 6 7
2. 反转整个数组: 7 6 5 4 3 2 1
3. 反转前k个元素 5 6 7 4 3 2 1
4. 反转其余元素 5 6 7 1 2 3 4

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

作者: sakura7301

链接: <https://leetcode-cn.com/problems/rotate-array/solution/reversefan-zhuan-san-ci-by-sakura7301-9ks7/>

来源: 力扣 (LeetCode)

著作权归作者所有。商业转载请联系作者获得授权, 非商业转载请注明出处。