



链滴

旋转数组

作者: [someone31851](#)

原文链接: <https://ld246.com/article/1546324460315>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

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。

示例 1:

****输入:**** [1,2,3,4,5,6,7] 和 $k = 3$

****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:**** [-1,-100,3,99] 和 $k = 2$

****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]

说明:

- 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。
- 要求使用空间复杂度为 $O(1)$ 的原地算法。

方法一:

```
class Solution {
    public void rotate(int[] nums, int k) {
        if (nums == null || nums.length == 0) {
            //return null;
        }
        int[] temp = new int[nums.length];
        for (int i = 0; i < k; i++) {

            temp[0] = nums[nums.length - 1];
            // System.out.println(temp[0]);
            for (int j = 1; j < nums.length; j++) {
                temp[j] = nums[j - 1];
            }

            //nums = temp;//将nums指向temp位置，不能这样写
            for (int x = 0; x < nums.length; x++) {
                nums[x] = temp[x];
            }
        }
    }
}
```

方法2:

...