



链滴

[每日 LeetCode] 189. Rotate Array

作者: [Hanseltu](#)

原文链接: <https://ld246.com/article/1552233229169>

来源网站: [链滴](#)

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

Description:

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

Example 1:

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

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

Explanation:

rotate 1 steps to the right: [7,1,2,3,4,5,6]

rotate 2 steps to the right: [6,7,1,2,3,4,5] rotate 3 steps to the right: [5,6,7,1,2,3,4]

Example 2:

Input: [-1,-100,3,99] and $k = 2$

Output: [3,99,-1,-100]

Explanation:

rotate 1 steps to the right: [99,-1,-100,3]

rotate 2 steps to the right: [3,99,-1,-100]

Note:

- Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.
- Could you do it in-place with $O(1)$ extra space?

思路：题目要求将数组旋转 k 次。可以先创建一个新数组，然后将原数组中的后 k 个元素放到新数组的 k 个位置，将原数组中的前 $n-k$ 个元素放到新数组的后 $n-k$ 个位置，然后将新数组的元素复制到旧数组，最后删除新数组。

注意：c++中memcpy函数用法

函数原型： `void *memcpy(void*dest, const void *src, size_t n);`

功能：由src指向地址为起始地址的连续n个字节的数据复制到以destin指向地址为起始地址的空间内。

C++代码

```
class Solution {
public:
    void rotate(vector<int>& nums, int k) {
        int n=nums.size();
        k=k%n;
        int* temp=new int[n];
        memcpy(temp,&nums[0]+(n-k),sizeof(int)*k);
        memcpy(temp+k,&nums[0],sizeof(int)*(n-k));
        memcpy(&nums[0],temp,sizeof(int)*n);
        delete[] temp;
    }
};
```

运行时间: 16ms

运行内存: