# 题目

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。

# 分析

class Solution {

public:

void rotate(vector<int>& nums, int k) {

//采用反转

int length=size(nums);

k%=size(nums);

reverse(&nums[0],&nums[length]);

reverse(&nums[0],&nums[k]);

reverse(&nums[k],&nums[length]);

}

};