

# LeetCode 第 239 号问题：滑动窗口最大值

本文首发于公众号「图解面试算法」，是 [图解 LeetCode](#) 系列文章之一。

同步博客：<https://www.algomooc.com>

题目来源于 LeetCode 上第 239 号问题：滑动窗口最大值。题目难度为 Hard，目前通过率为 40.5%。

## 题目描述

给定一个数组 *nums*，有一个大小为 *k* 的滑动窗口从数组的最左侧移动到数组的最右侧。你只可以看到在滑动窗口 *k* 内的数字。滑动窗口每次只向右移动一位。

返回滑动窗口最大值。

## 示例:

输入: *nums* = [1,3,-1,-3,5,3,6,7], 和 *k* = 3

输出: [3,3,5,5,6,7]

解释:

滑动窗口的位置	最大值
-----	-----
[1 3 -1] -3 5 3 6 7	3
1 [3 -1 -3] 5 3 6 7	3
1 3 [-1 -3 5] 3 6 7	5
1 3 -1 [-3 5 3] 6 7	5
1 3 -1 -3 [5 3 6] 7	6
1 3 -1 -3 5 [3 6 7]	7

## 注意:

你可以假设 *k* 总是有效的， $1 \leq k \leq$  输入数组的大小，且输入数组不为空。

## 进阶:

你能在线性时间复杂度内解决此题吗？

## 题目解析

利用一个 **双端队列**，在队列中存储元素在数组中的位置，并且维持队列的严格递减，也就说维持队首元素是 \*\*最大的 \*\*，当遍历到一个新元素时，如果队列里有比当前元素小的，就将其移除队列，以保证队列的递减。当队列元素位置之差大于 *k*，就将队首元素移除。

## 补充：什么是双端队列（Deque）

Deque 的含义是 “double ended queue”，即双端队列，它具有队列和栈的性质的数据结构。顾名思义，它是一种前端与后端都支持插入和删除操作的队列。

Deque 继承自 Queue（队列），它的直接实现有 ArrayDeque、LinkedList 等。

## 动画描述

## 代码实现

```
class Solution {
    public int[] maxSlidingWindow(int[] nums, int k) {
        //有点坑，题目里都说了数组不为空，且 k > 0。但是看了一下，测试用例里面还是有nums = [], k
        = 0，所以只好加上这个判断
        if (nums == null || nums.length < k || k == 0) return new int[0];
        int[] res = new int[nums.length - k + 1];
        //双端队列
        Deque<Integer> deque = new LinkedList<>();
        for (int i = 0; i < nums.length; i++) {
            //在尾部添加元素，并保证左边元素都比尾部大
            while (!deque.isEmpty() && nums[deque.getLast()] < nums[i]) {
                deque.removeLast();
            }
            deque.addLast(i);
            //在头部移除元素
            if (deque.getFirst() == i - k) {
                deque.removeFirst();
            }
            //输出结果
            if (i >= k - 1) {
                res[i - k + 1] = nums[deque.getFirst()];
            }
        }
        return res;
    }
}
```