画解剑指 Offer - LeetBook - 力扣（LeetCode）全球极客挚爱的技术成长平台

剑指 Offer 59 - I. 滑动窗口的最大值 - 解决方案

题目描述

给定一个数组 `nums` 和滑动窗口的大小 `k` ，请找出所有滑动窗口里的最大值。

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
```

提示：

你可以假设 k 总是有效的，在输入数组不为空的情况下，1 ≤ k ≤ 输入数组的大小。

解题方案

思路

- 标签：单调队列
- 整体思路：
  - 从题目上来看是通过维护滑动窗口，然后每次求滑动窗口中的最大值即可，设数组长度为 n，窗口长度为 k，则时间复杂度为 O(k*(n-k+1)) = O(kn)O(k∗(n−k+1))=O(kn)
  - 很显然使用暴力解法的话，时间复杂度会随着 k 变大不断变大，而其中有很多元素在不同的滑动窗口中都存在着，所以必然存在重复计算的逻辑
  - 考虑使用单调队列，队列内只存在窗口内的元素，队列内元素递减。可以保证所有的数据只会入队和出队一次，减少时间复杂度

- 复杂度：
  - 时间复杂度：O(n)O(n)。遍历数组需要 O(n)O(n) 的时间复杂度，数组中的元素最多入队和出队一次，队列内元素维护最多需要 O(2n)O(2n) ，所以总体时间复杂度为 O(n)O(n)
  - 空间复杂度：O(k)O(k)。维护一个最多元素个数为 k 个的队列

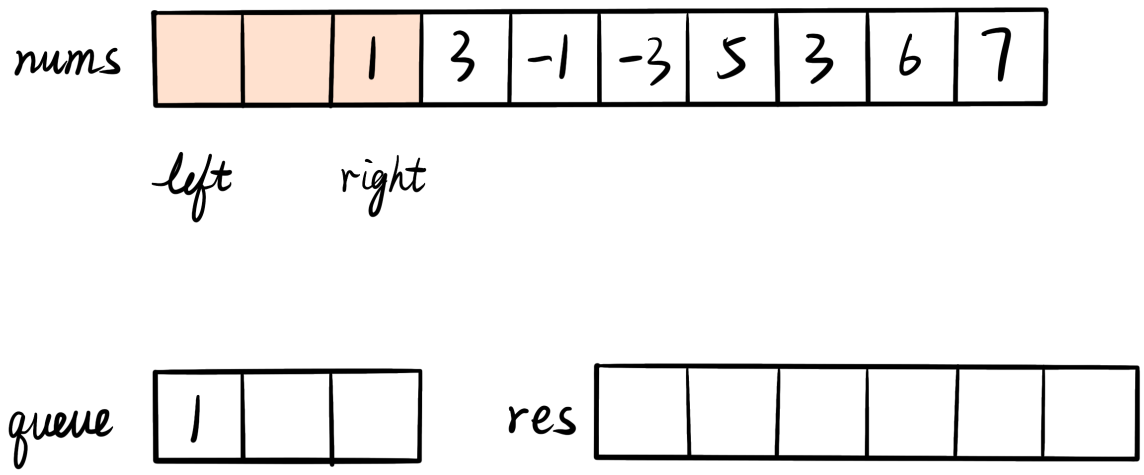
算法流程

1. 初始化滑动窗口的 left 和 right 位置，从下标为 [1-k, 0] 范围开始
2. 如果 left > 0 说明窗口已经在数组中了，并且单调队列的第一个元素和 nums[left - 1] 相等时，说明该元素已经不在滑动窗口中，需要移除
3. 如果单调队列不为空且最后一个元素小于新加入的 nums[right] 元素，则需要维护单调队列为递减状态，所以将最后一个元素移除，直到其大于新加入元素
4. 将新加入的 nums[right] 元素加入单调队列，因为上一步的操作，当前单调队列一定是递减的
5. 如果 left >= 0，说明窗口在数组中，因为单调队列递减，所以第一个元素一定是当前滑动窗口最大值

代码

```
class Solution {
    public int[] maxSlidingWindow(int[] nums, int k) {
        if(nums.length == 0 || k == 0) {
            return new int[0];
        }
        Deque<Integer> queue = new LinkedList<>();
        int[] res = new int[nums.length - k + 1];
        for(int right = 0, left = 1 - k; right < nums.length; left++, right++) {
            if(left > 0 && queue.peekFirst() == nums[left - 1]) {
                queue.removeFirst();
            }
            while(!queue.isEmpty() && queue.peekLast() < nums[right]) {
                queue.removeLast();
            }
            queue.addLast(nums[right]);
            if(left >= 0) {
                res[left] = queue.peekFirst();
            }
        }
        return res;
    }
}
```

画解



1 / 9

花絮

讨论区
共 2 个讨论

最热

admin L1
来自湖南2021-12-12

这个单调队列可以类比成生活中的插队。

一大哥来排队，看见最后一个人是弱鸡，就直接插他前面。以此类推，这位大哥刚好站在他打不过的人后边。当然有个前提，所有来排队的人都是这种恃强凌弱的方式排队。

这窗口又是怎么回事呢？

窗口右边是刚来排队的人，窗口左边是刚走的人，这人可能是刚办好事，也可能是当前排队的大哥都打不过。最后把办过事的人的列表输出。

2

笨熊 L5
来自广东2021-07-04

。。