

剑指 Offer 59 - I. 滑动窗口的最大值

Friday, June 17, 2022 9:46 AM

<https://leetcode.cn/problems/hua-dong-chuang-kou-de-zui-da-zhi-lcof/>

困难

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

我的思路，for循环遍历`nums`，内嵌`k`长度的for循环。 $O(nk)$ $O(k)$

优化解法 $O(n)$ $O(k)$: 一个双端队列维护窗口下标，队首为窗口内最大值下标，队首后面值为队首下标后面的下标，且对应值小于队首下标对应值

步骤:

设一个队列，队列维护`k`长度内的下标

队空时直接加入当前下标，非空时，while比较当前值与队列尾值对应的`nums`值

若大与等于，则删除队列尾值，继续往前比较删除，

若小于，不操作

队尾加入当前下标。

因为 以上操作，保证了队头的值为下标最小值，

故在遍历时判断其是否已出窗口，进行删除即可保证队内元素为窗口内的

例子: 123214, 2

队列值:	对应nums值	操作
0	1	队空，下标直接入队
1	2	对应nums值大，删除队尾，下标入队
2	3	对应nums值大，删除队尾，下标入队
23	32	对应nums值小，直接入队
34	21	对应nums值小，队首值已脱离窗口，队首出队，下标入队
5	4	对应nums值大，循环比较删除队尾，下标入队

```

class Solution{
    public int[] maxSlidingWindow(int[] nums, int k) {
        if(nums.length == 0 || k == 0) return new int[0];
        int n = nums.length;
        Deque<Integer> deque = new LinkedList<>();
        // 初始化队列，队首为第一个窗口最大值下标
        for (int i = 0; i < k; ++i) {
            // 队列非空时，且 nums[i] 大于等于队列中最后一个下标对应nums值时，
            // 删除该下标
            // 即：新值比之前大或等于，就删了前面的，入队
            // 新值小就直接入队
            while (!deque.isEmpty() && nums[i] >= nums[deque.peekLast()]) {
                deque.pollLast();
            }
            // 下标 i 插入队尾。
            deque.offerLast(i);
        }
        int[] ans = new int[n - k + 1];
        // 第一个窗口最大值，此时已可直接拿到
        ans[0] = nums[deque.peekFirst()];
        // 继续遍历 现在起要加入队首的下标越界判断
        for (int i = k; i < n; ++i) {
            // 新值加入策略不变
            while (!deque.isEmpty() && nums[i] >= nums[deque.peekLast()]) {
                deque.pollLast();
            }
            deque.offerLast(i);
            // 判断队首是否不在窗口内了，是就删了
            while (deque.peekFirst() <= i - k) {
                deque.pollFirst();
            }
            // 每轮取一次窗口最大值
            ans[i - k + 1] = nums[deque.peekFirst()];
        }
        return ans;
    }
}

```