0239. 滑动窗口最大值

ITCharge 大约 2 分钟

- 标签：队列、数组、滑动窗口、单调队列、堆（优先队列）
- 难度：困难

题目链接

- 0239. 滑动窗口最大值 - 力扣

题目大意

描述：给定一个整数数组 `nums` ，再给定一个整数 `k` ，表示为大小为 `k` 的滑动窗口从数组的最左侧移动到数组的最右侧。我们只能看到滑动窗口内的 `k` 个数字，滑动窗口每次只能向右移动一位。

要求：返回滑动窗口中的最大值。

说明：

- $1 \leq nums.length \leq 10^5$。
- $-10^4 \leq nums[i] \leq 10^4$。
- $1 \leq k \leq nums.length$。

示例：

- 示例 1：

```py
输入：nums = [1,3,-1,-3,5,3,6,7], 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
```

- 示例 2：

```py
输入：nums = [1], k = 1
输出：[1]
```

解题思路

暴力求解的话，需要使用二重循环遍历，其时间复杂度为 $O(n * k)$。根据题目给定的数据范围，肯定会超时。

我们可以使用优先队列来做。

思路 1：优先队列

1. 初始的时候将前 `k` 个元素加入优先队列的二叉堆中。存入优先队列的是数组值与索引构成的元组。优先队列将数组值作为优先级。
2. 然后滑动窗口从第 `k` 个元素开始遍历，将当前数组值和索引的元组插入到二叉堆中。
3. 当二叉堆堆顶元素的索引已经不在滑动窗口的范围中时，即 `q[0][1] <= i - k` 时，不断删除堆顶元素，直到最大值元素的索引在滑动窗口的范围中。
4. 将最大值加入到答案数组中，继续向右滑动。
5. 滑动结束时，输出答案数组。

思路 1：代码

```py
class Solution:
    def maxSlidingWindow(self, nums: List[int], k: int) -> List[int]:
        size = len(nums)
        q = [(-nums[i], i) for i in range(k)]
        heapq.heapify(q)
        res = [-q[0][0]]
```

```
        for i in range(k, size):
            heapq.heappush(q, (-nums[i], i))
            while q[0][1] <= i - k:
                heapq.heappop(q)
            res.append(-q[0][0])
        return res
```

思路 1：复杂度分析

- **时间复杂度**：$O(n \times \log_2 n)$。
- **空间复杂度**：$O(k)$。