# 59. 滑动窗口的最大值

## 题目链接

[牛客网](https://www.nowcoder.com/practice/1624bc35a45c42c0bc17d17fa0cba788?tpId=13&tqId=11217&tPage=1&rp=1&ru=/ta/coding-interviews&qru=/ta/coding-interviews/question-ranking&from=cyc_github)

## 题目描述

给定一个数组和滑动窗口的大小，找出所有滑动窗口里数值的最大值。

例如，如果输入数组 {2, 3, 4, 2, 6, 2, 5, 1} 及滑动窗口的大小 3，那么一共存在 6 个滑动窗口，他们的最大值分别为 {4, 4, 6, 6, 6, 5}。

## 解题思路

维护一个大小为窗口大小的大顶堆，顶堆元素则为当前窗口的最大值。

假设窗口的大小为 M，数组的长度为 N。在窗口向右移动时，需要先在堆中删除离开窗口的元素，并将新到达的元素添加到堆中，这两个操作的时间复杂度都为 log2M，因此算法的时间复杂度为 O(Nlog2M)，空间复杂度为 O(M)。

public ArrayList<Integer> maxInWindows(int[] num, int size) {
 ArrayList<Integer> ret = new ArrayList<>();
 if (size > num.length || size < 1)
 return ret;
 PriorityQueue<Integer> heap = new PriorityQueue<>((o1, o2) -> o2 - o1); /\* 大顶堆 \*/
 for (int i = 0; i < size; i++)
 heap.add(num[i]);
 ret.add(heap.peek());
 for (int i = 0, j = i + size; j < num.length; i++, j++) { /\* 维护一个大小为 size 的大顶堆 \*/
 heap.remove(num[i]);
 heap.add(num[j]);
 ret.add(heap.peek());
 }
 return ret;
}