# 题目

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内的 k 个数字。滑动窗口每次只向右移动一位。

返回滑动窗口中的最大值。

**进阶：**

你能在线性时间复杂度内解决此题吗？

**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：**

1 <= nums.length <= 10^5

-10^4 <= nums[i] <= 10^4

1 <= k <= nums.length

# 分析

## 方法一：暴力破解

## 方法二：双向队列

**思路：**

**代码：**

class Solution {

public:

vector<int> maxSlidingWindow(vector<int>& nums, int k) {

if(0==nums.size())return {};

vector<int> ret;

deque<int> deq; //存储最大值的下标

for(int i=0;i<nums.size();i++)

{

while(!deq.empty() && nums.at(i) > nums.at(deq.back()))

deq.pop\_back();

if(!deq.empty() && deq.front()<i-k+1)deq.pop\_front();

deq.push\_back(i);

if(i>=k-1)ret.push\_back(nums.at(deq.front()));

}

return ret;

}

};

## 方法三：动态规划