一、题目说明

题目239. Sliding Window Maximum，给一个数组，和窗口长度，窗口每次向右滑动1位，返回滑动窗口的最大值。时间复杂度要求是线性。

二、我的解答

最直接的办法就是`brute force`，但是性能不足，复杂度是O(kN)：

```
class Solution{
    public:
        //brute force
        vector<int> maxSlidingWindow(vector<int>& nums, int k){
            int start = 0,curMax=0;
            vector<int> result;
            while(start+k <= nums.size()){
                curMax = nums[start];
                for(int i=start;i<start+k;i++){
                    if(nums[i]>curMax) curMax = nums[i];
                }
                result.push_back(curMax);
                start++;
            }

            return result;
        }
};
```

性能如下：

```
Runtime: 100 ms, faster than 16.44% of C++ online submissions for Sliding Window Maximum.
Memory Usage: 13 MB, less than 86.89% of C++ online submissions for Sliding Window Maximum.
```

三、优化措施

用双端队列实现：

```
class Solution{
    public:
        //use deque to store index,window.front() to store the max value of current window
        //nums[i],将双端队列deque从尾部开始把每个小于nums[i]的数去除。
        //nums[i-k]等于deque头的元素，要把deque的头去除掉，相当于一个最大值移出了窗口
        vector<int> maxSlidingWindow(vector<int>& nums, int k){
            vector<int> result;
            if(k==0 || nums.size()<1) return result;
            deque<int> window;
            //init the first k nums
            for(int i=0;i<k;i++){
                while(! window.empty() && nums[i]>nums[window.back()]){
                    window.pop_back();
```

```
                }
                window.push_back(i);
            }
            result.push_back(nums[window.front()]);

            //other data
            for(int i=k;i<nums.size();i++){
                if(!window.empty() && window.front()<=i-k){
                    window.pop_front();
                }
                while(!window.empty() && nums[i]>nums[window.back()]){
                    window.pop_back();
                }
                window.push_back(i);
                result.push_back(nums[window.front()]);
            }
            return result;
        }
};
```

性能如下：

```
Runtime: 60 ms, faster than 63.43% of C++ online submissions for Sliding Window Maximum.
Memory Usage: 13.3 MB, less than 59.02% of C++ online submissions for Sliding Window Maximum.
```