

题目描述:

给定数组 $nums$, 有大小为 k 的滑动窗口从数组最左侧移动到最右侧, 返回滑动窗口的最大值

示例:

输入: $nums = [1, 3, -1, -3, 5, 3, 6, 7]$ $k = 3$

$[1, 3, -1]$ $-3, 5, 3, 6, 7$ 3

$[3, -1, -3]$ $5, 3, 6, 7$ 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

思路:

用队列保存滑动窗口里的值, 给出最大的值

单调队列:

维护一个队列单调增或单调减

模拟过程:

因为我们只需要最大的元素, 加入单调队列时, 在某元素前比其还要小的元素可以遗弃掉,

如果, 加入单调队列时, 比前面元素小, 直接加入单调队列

代码实现:

```
class MyQueue
```

```
    deque<int> que
```

```
    pop(int value) {
```

```
        if (!que.empty() && value == que.front())
```

```
            que.pop_front();
```

```
    }
```

```
    push(int value) {
```

```
        while (!que.empty() && value > que.back())
```

```
            que.pop_back();
```

```
        que.push_back(value)
```

```
    }
```

```
    get_max_value() {
```

```
        return que.front();
```

```
    }
```

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

```
    MyQueue que;
```

```
    vector<int> result;
```

```
    for (int i=0; i<k; i++) {
```

```
        que.push(nums[i]);
```

```
    }
```

```
    result.push_back(que.front());
```

```
    for (int i=k; i<nums.size(); i++) {
```

```
        que.pop(nums[i-k]);
```

```
        que.push(nums[i]);
```

```
        result.push(que.front());
```

```
    }
```

```
    return result;
```

```
}
```