

滑动窗口的最大值

给定一个数组 `nums` 和滑动窗口的大小 `k`，请找出所有滑动窗口里的最大值。

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

```
class Solution {
public:
    vector<int> maxSlidingWindow(vector<int>& nums, int k) {
        vector<int> res;
        if(nums.size()==0)
        {
            return res;
        }
        int n=nums.size()-k+1;//窗口数
        int t=-1;//当前窗口最大值下标
        int max;//当前窗口最大值

        for(int i=0;i<n;i++)
        {
            if(t<i)//上个滑窗最大值不在当前窗口里 重新遍历查找
            {
                max=nums[i];
                for(int j=i+1;j<i+k;j++)
                {
                    if(nums[j]>max)
```

```
        {
            t=j;
            max=nums[j];
        }
    }
    else//上个滑窗最大值在当前窗口里
    {
        if(nums[i+k-1] >= max)//注意，此时滑动窗口移动，只新增一个
数 只需要与新增的数比较即可
        {
            t = i+k-1;
            max = nums[i+k-1];
        }
        res.push_back(max);
    }
    return res;
}
};
```