배열 nums가 주어졌을 때 k 크기의 슬라이딩 윈도우를 오르쪽 끝까지 이동하면서 최대 슬라이딩 윈도우를 구하라.

Input: nums = [1,3,-1,-3,5,3,6,7], k = 3
Output: [3,3,5,5,6,7]
Explanation:

```
Window position                Max
---------------               -----
[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
```

1.deque 활용

```
from collections import deque

class Solution:
    def maxSlidingWindow(self, nums: List[int], k: int) -> List[int]:
        q = deque()
        result = []
        for i in range(len(nums)):
            # the first/left (max) element is out of the current window
            if q and i - q[0] == k:
                q.popleft()
            while q:
                # pop useles elements from last/right of the queue
                if nums[q[-1]] < nums[i]:
                    q.pop()
                else:
                    break
            q.append(i)
            if i >= k-1: # i == k-1 is the beginning of a full window
                result.append(nums[q[0]])
        return result
```