

239. Sliding Window Maximum

Hard 6510 246 Add to List Share

You are given an array of integers `nums`, there is a sliding window of size `k` which is moving from the very left of the array to the very right. You can only see the `k` numbers in the window. Each time the sliding window moves right by one position.

Return the max sliding window.

Example 1:

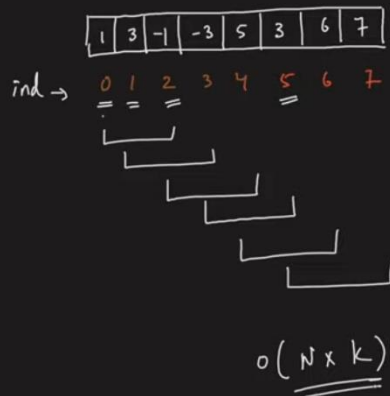
Input: `nums = [1,3,-1,-3,5,3,6,7]`, `k = 3`
Output: `[3,3,5,5,6,7]`

Explanation:

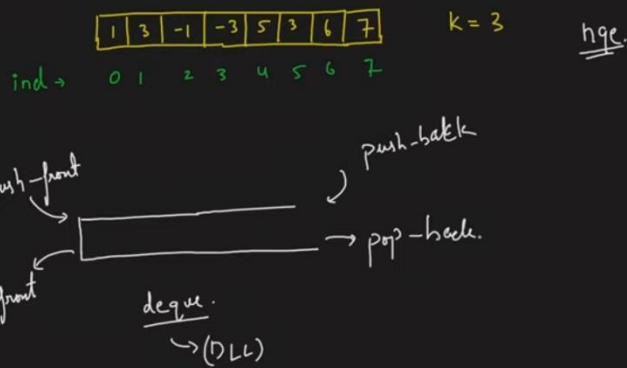
Window position	Max
[1 3 -1] -3 5 3 6 7	3
1 [3 -1 -3] 5 3 6 7	3
1 3 [-1 -3 5] 3 6 7	5
1 3 -1 [-3 5 3] 6 7	5
1 3 -1 -3 [5 3 6] 7	6
1 3 -1 -3 5 [3 6 7]	7

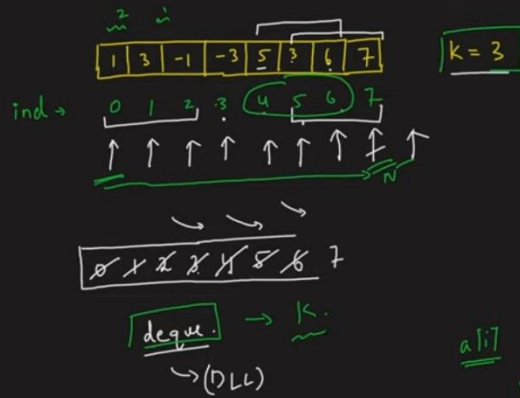
Example 2:

Input: `nums = [1]`, `k = 1`



for ($i = 0 \rightarrow 5$) $\forall N$
 $maxi = a[j]$
 for ($j = i \rightarrow i + k - 1$) $\forall k$
 $maxi = \max(a[j], maxi)$
 $\}$





TC $\rightarrow O(N) + O(N) \approx O(N)$

SC $\rightarrow O(k)$

all \rightarrow out of bound
 $\rightarrow \leq a[i]$

TUF

```

1 class Solution {
2 public:
3     vector<int> maxSlidingWindow(vector<int>& nums, int k) {
4         deque<int> dq;
5         vector<int> ans;
6         for (int i=0; i<nums.size(); i++) {
7             if (!dq.empty() && dq.front() == i-k) dq.pop_front();
8
9             while (!dq.empty() && nums[dq.back()] < nums[i])
10                 dq.pop_back();
11
12             dq.push_back(i);
13             if (i>=k-1) ans.push_back(nums[dq.front()]);
14         }
15         return ans;
16     }
17 };

```

Your previous code was restored from your local storage. [Reset to default](#)

TUF

```

1 class Solution {
2 public:
3     vector<int> maxSlidingWindow(vector<int>& nums, int k) {
4         deque<int> dq;
5         vector<int> ans;
6         for (int i=0; i<nums.size(); i++) {
7             if (!dq.empty() && dq.front() == i-k) dq.pop_front();
8
9             while (!dq.empty() && nums[dq.back()] <= nums[i])
10                 dq.pop_back();
11
12             dq.push_back(i);
13             if (i>=k-1) ans.push_back(nums[dq.front()]);
14         }
15         return ans;
16     }
17 };

```

Handwritten annotations in pink:

- Diagram showing a window of size $k=3$ at index i .
- Annotation: \rightarrow out of bound (pointing to $i-k$ in line 7).
- Annotation: \rightarrow remove $\leq a[i]$ (pointing to the while loop in line 9).
- Diagram showing the array $[0, 1, 2]$ with $k=3$ and an arrow pointing to index i .

Your previous code was restored from your local storage. [Reset to default](#)

TUF