

* Sliding Window Maximum

[1, 3, -1, -3, 5, 3, 6, 7] size = 3

The diagram illustrates a sliding window of size 3 moving across the array [1, 3, -1, -3, 5, 3, 6, 7]. The window starts at index 0 and moves to index 7. The elements in the window are underlined: [1, 3, -1], [3, -1, -3], [-1, -3, 5], [-3, 5, 3], [5, 3, 6], [3, 6, 7].

use priority Queue \rightarrow maxQueue.

- use a double linked List (Deque)
LinkedList / Array Deque

Keep decreasing sequence of values.

If index of top value / first value is out of window, pop it. When add a value, maintain the order by popping from end.

1 3 -1 -3 5 3 6 7

(m)

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

1 2 3 4 5 6 7

• use 2 arrays & in 2 traversals fill both ; left to right & right to left.

if $i \% k = 0$, put the same value.

Else take the max of

3 -1 -3 5 3 6 7

1 3 3 -3 5 5 6 7

previous & current.

right

3 3 -1 5 5 3 7 7

if $(i+1) \% k = 0$ put the same value. Else put the max of current & right.

for each place take the max of right, left relevant value.