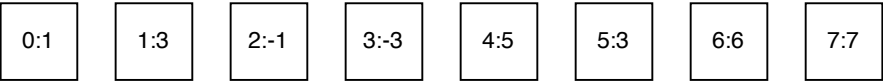


Boxes representing entries in the array contains its index and value like the following:



Example array



Window size k = 3

Solution

Let a double ended queue represent the array entries within the sliding window

Store the indices of the elements in the deque

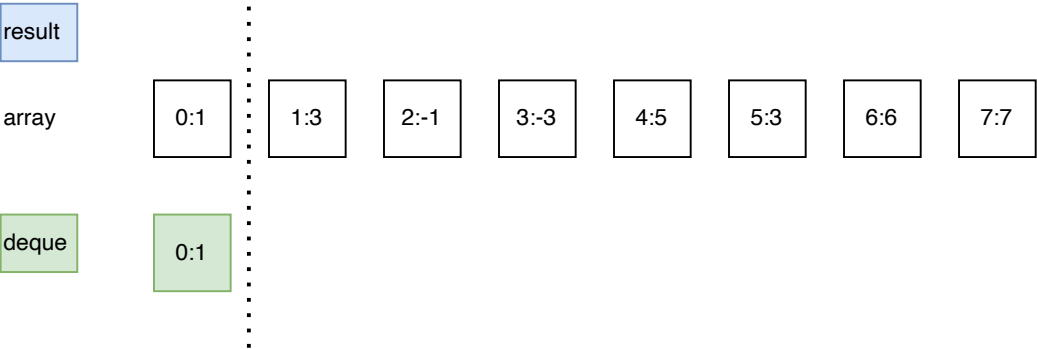
Largest element in the deque shall always be at the top

Loop through the array item by item

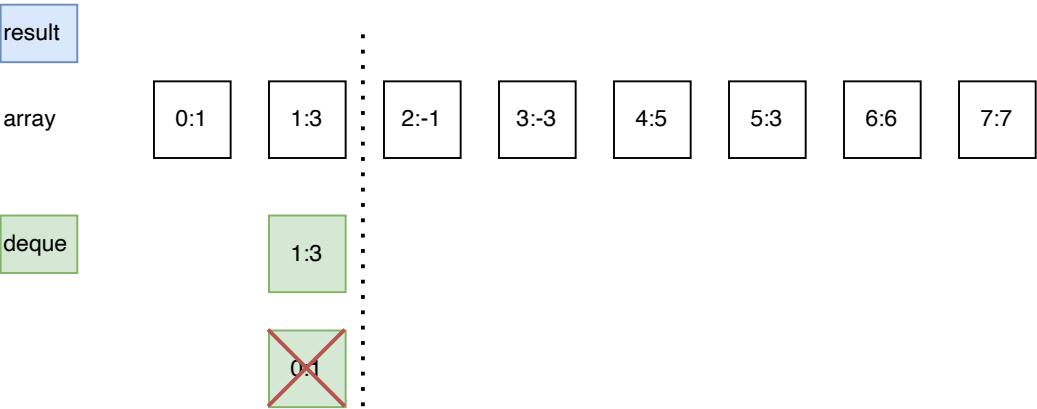
For each item

1. Remove element at the top if it is outside of the window
2. Maintain the deque so that all items in the queue that is smaller than the current item is removed, they cannot contribute
3. Add the current item to the back of the queue since all smaller items has been removed in step 2
4. If current item is not within the first k-1 entries add the top element if the deque to the result array

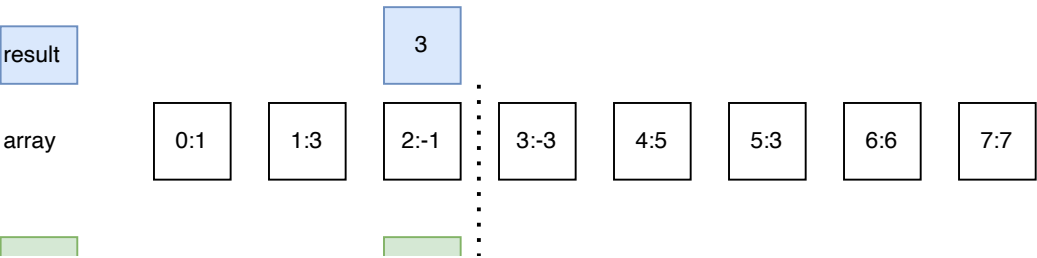
i = 0



i = 1



i = 2



deque

1:3

2:-1

i = 3

result

3

3

array

0:1

1:3

2:-1

3:-3

4:5

5:3

6:6

7:7

deque

1:3

2:-1

3:-3

i = 4

result

3

3

5

array

0:1

1:3

2:-1

3:-3

4:5

5:3

6:6

7:7

deque

4:5

not in window

1:3

smaller than  
current

2:-1

smaller than  
current

3:-3

i = 5

result

3

3

5

5

array

0:1

1:3

2:-1

3:-3

4:5

5:3

6:6

7:7

deque

4:5

5:3

i = 6

result

3	3	5	5	6
---	---	---	---	---

array

0:1	1:3	2:-1	3:-3	4:5	5:3	6:6	7:7
-----	-----	------	------	-----	-----	-----	-----

deque

6:6

smaller than  
current

~~4:5~~

smaller than  
current

~~5:3~~

i = 7

result

3	3	5	5	6	7
---	---	---	---	---	---

array

0:1	1:3	2:-1	3:-3	4:5	5:3	6:6	7:7
-----	-----	------	------	-----	-----	-----	-----

deque

7:7

smaller than  
current

~~6:6~~