

⑧ Sliding window Maximum

(K sized subarray Maximum)

Example

Input: $K=3$, $arr[] = [1, 2, 3, 1, 4, 5, 2, 3, 6]$

o/p: $[3, 3, 4, 5, 5, 5, 6]$

Explanation: $[1, 2, 3]$ $\max=3$

$[2, 3, 1]$ $\max=3$

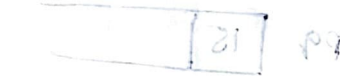
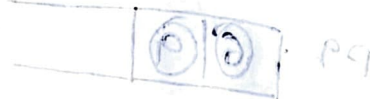
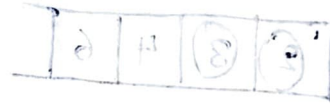
$[3, 1, 4]$ $\max=4$

$[1, 4, 5]$ $\max=5$

$[4, 5, 2]$ $\max=5$

$[5, 2, 3]$ $\max=5$

$[2, 3, 6]$ $\max=6$



Approach:

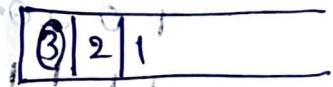


— at max K elements.

use ArrayList to store



① add K nums to pq (1st window)



② $i=0$ $pq.peek() \rightarrow arr[i]$

~~while~~ while ($pq.peek().idx \leq (i-K)$) {

$pq.remove()$;

}

$pq.add(curr)$

$wind[i] = pq.peek()$.

$K=3$

4	7	5	2	1
0	1	2	3	4

Idx

0	1	2	3	4
--------------	--------------	---	---	---

ele

4	7	5	2	1
--------------	--------------	---	---	---

Res = [7, 7, 5] — return

$K=3$

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

Res = [3, 3, 5, 5, 6, 7]

Idx

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

ele

1	3	-1	-3	5	3	6	7
--------------	--------------	---------------	---------------	---	---	--------------	---