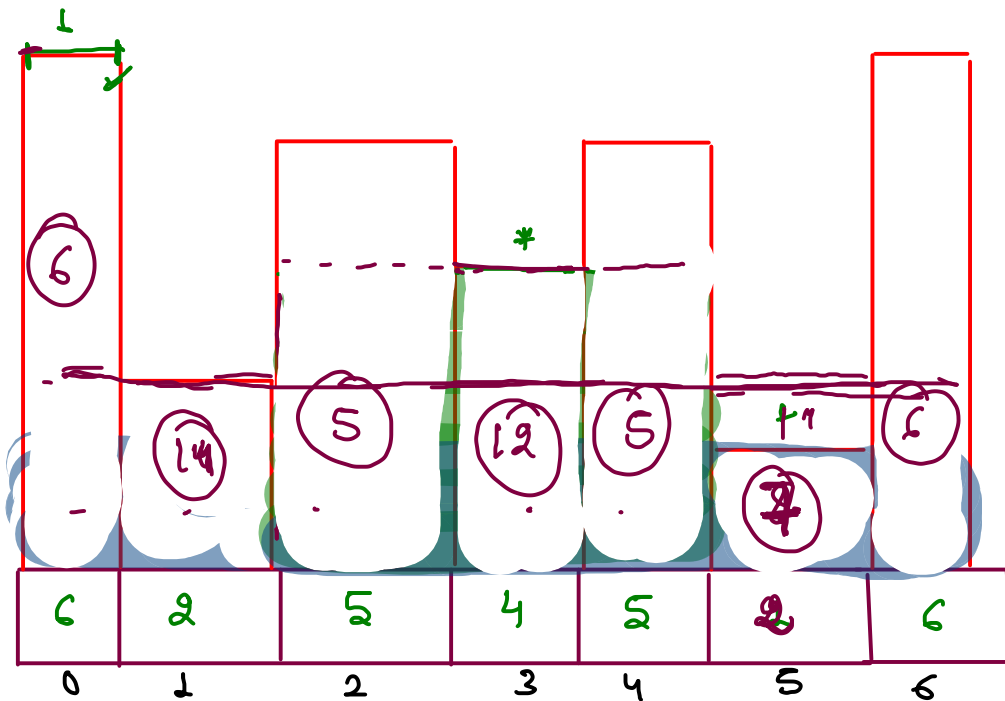


Longest Area Histogram:

- * width of each bar = 1
- * height is given in array



* Area of Largest Rectangle.

$$\text{Area} = \underbrace{(rsi - lsi - 1)}_{\text{width}} * \text{height}$$

$$5 - (-1) = 6$$

left smaller index (lsi) →

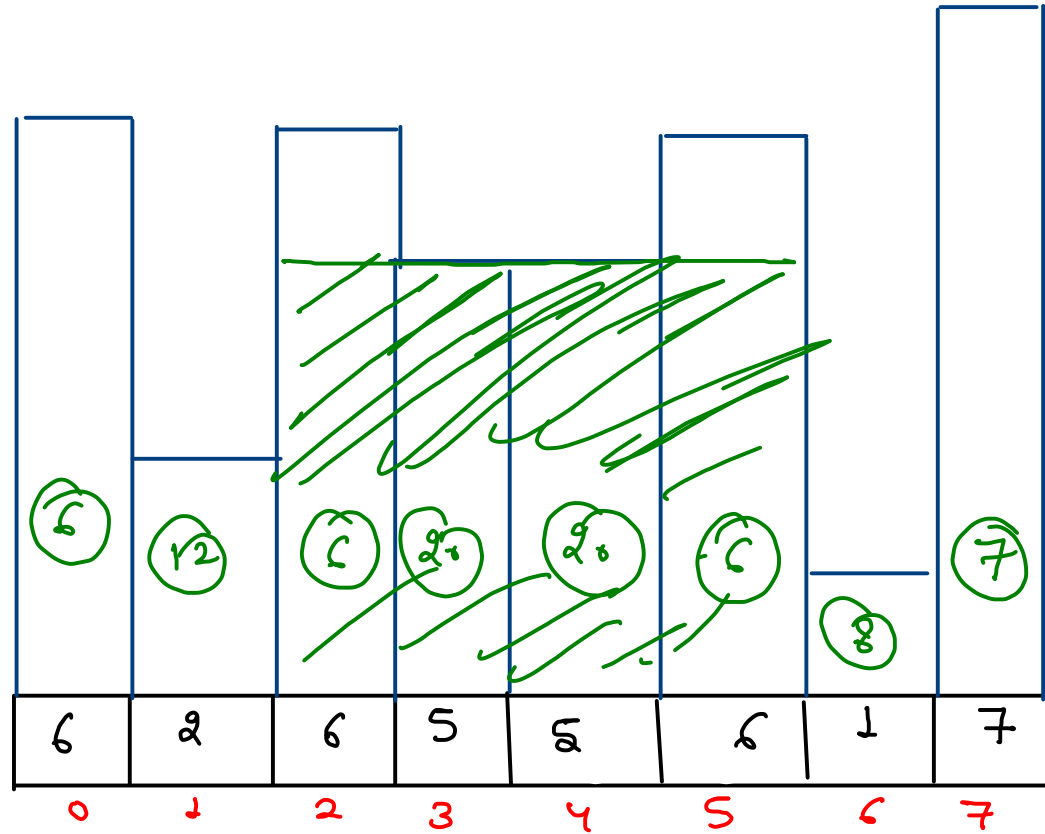
-1	-1	1	1	2	-1	5
----	----	---	---	---	----	---

right smaller index (rsi) →

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

→ By default = -1

→ By default = arr.length



✓ Right Smaller Index →

(rsi - lsi - 1) →

✓ left Smaller Index →

1	6	2	5	5	5	8	8
1	6	2	4	9	1	8	1
-1	-1	1	1	1	4	-1	6

width

Maximal Rectangle

largest Area
Histogram

1	0	1	0	0
1	0	1	1	1
1	1	1	1	1
1	0	0	1	0

1	0	1	0	0
---	---	---	---	---

2	0	2	1	1
---	---	---	---	---

2	1	2	2	2
---	---	---	---	---

4	0	0	2	0
---	---	---	---	---

1

3

5

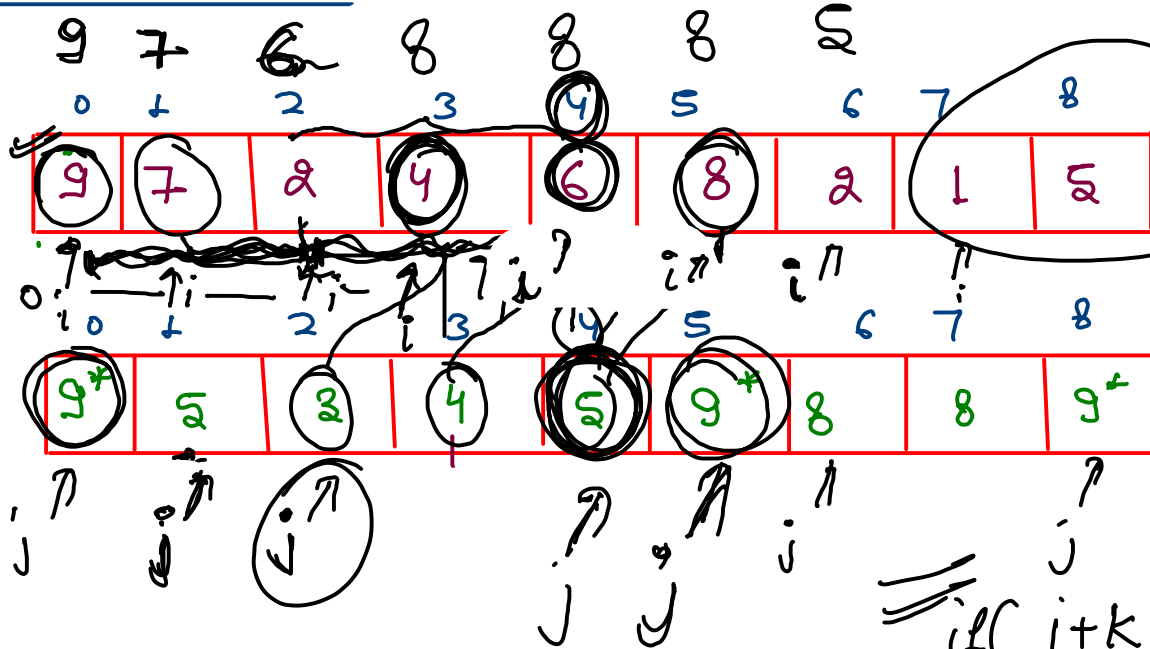
4

max = 5

Sliding Window Maximum.

$k=3$

Next Greater Index on Right



if (i > j) {
j = i;

if (i+k > nge[j]) {

}

j = 0;

(i = 0; arr.length - 1; i++) {
 if (i > j) {

j = i;
 }

while (i < k ~~arr~~ arr[i]) {

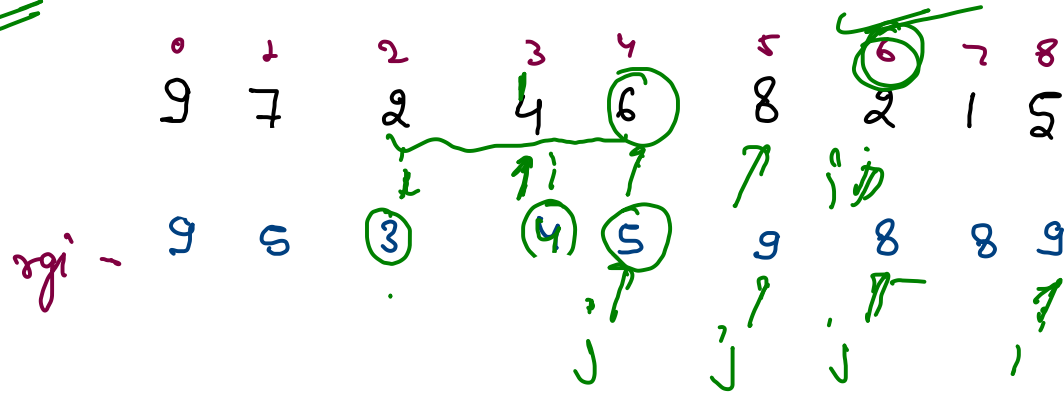
j = arr[i];

}
 swap(arr[i],

arr[i] + right
greater index

k = 3

k=3



3+3 =
⑤

4+3 > 5

i+k = 2+3 = ⑤

5 > ④

5 > 5

9
7
6
8
8
8
5

```
public static void slidingWindowMax(int[] arr, int k) {
    int[] rgi = rightGreaterIndex(arr);

    int j = 0;
    for(int i = 0; i < arr.length - k; i++) {
        if(i > j) {
            j = i;
        }
        while(i + k < rgi[j]) {
            j = rgi[j];
        }
        System.out.println(arr[j]);
    }
}
```

9-k = 9-3 = ⑥