https://course.acciojob.com/idle?question=2da7ad22-cccc-497d-864c-a3ea784e1263

- MEDIUM
- Max Score: 40 Points
-
-
-
-
-

Sliding window Maximum

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 maximum of each sliding window.

Complete the given function `SlidingWindowMaximum` which receives the input array, `n` and `k` as its parameters and returns an array containing maximum of each window of size `k`.

Input Format

The first line contains `N` and `K` denoting the number of elements in the array/list and value of `k`.

The second line contains `N` single space-separated integers denoting the elements of the array.

NOTE: You do not need to print anything; it has already been taken care of.

Just Complete the function.

Output Format

Print the max sliding window.

Example 1

Input

```
1 1
1
```

Output:

```
1
```

Explanation:

Maximum window will be 1.

Example 2

Input

```
8 3
1 3 -1 -3 5 3 6 7
```

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
```

Constraints

1 <= `N` <= 20000

1 <= `K` <= `N`

-10^4 <= `arr[i]` <= 10^4

Topic Tags

- **Queues**
- **Deques**
- **Sliding Window**
- **Arrays**

My code

```
// n java
import java.util.*;

class Solution{
    static int[] SlidingWindowMaximum(int n, int k, int[] arr){
        // write code here
                int ar[]=new int[n-k+1];
                for(int i=0;i<=n-k;i++)
                        {
                                int max=arr[i];
                                for(int j=i;j<(i+k);j++)
                                        if(arr[j]>max)
                                                max=arr[j];
                                 ar[i]=max;
                        }
                return ar;
    }
}

public class Main {
    public static void main(String[] args) throws Throwable {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int k = sc.nextInt();
        int nums[]=new int[n];
        for(int i = 0; i < n; i++)
        {
            nums[i] = sc.nextInt();
        }
        Solution obj = new Solution();
        int[] ans=obj.SlidingWindowMaximum(n,k,nums);
        for(int i=0;i<ans.length;++i){
            System.out.print(ans[i] + " ");
        }
    }
}
```