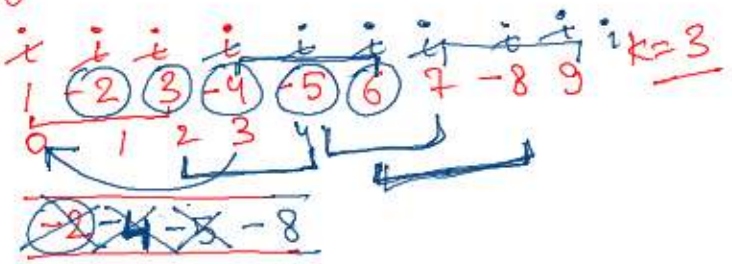
Qu — first Negative — sliding window

```
    Queue<Integer> que = new ArrayDeque();

    for(int i=0;i<k;i++){
        if(arr[i]<0){
            que.offer(arr[i]);
        }
    }

    if(!que.isEmpty()){
        System.out.print(que.peek()+" ");
    }else{
        System.out.print("0"+" ");
    }

    for(int i=k;i<arr.length;i++){
        if(!que.isEmpty() && que.peek()==arr[i-k]){// to remove element outside window
            que.poll();
        }
        if(arr[i]<0){//add only -ve value
            que.offer(arr[i]);
        }
        if(!que.isEmpty()){// print first -ve value of window
            System.out.print(que.peek()+" ");
        }else{
            System.out.print("0"+" ");
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be n
}
```



[-2 -2 -4 -4 -5 -8 -8]

1. push all -ve value of k slide
2. print first -ve value for that slide
3. pop -ve value outside from my bucket
4. add value to queue if -ve.
5. print peek(); ? 0;

Priority Queue / Heap

[0 , -3 , 5 , -2 , 8 , 9]

1) min pq

2) max pq

3 { pq.poll()
  .remove()

pq

pq.poll() } -3
pq.remove()

min pq

Syntax

PriorityQueue<Integer> pq = new PriorityQueue<>();

PriorityQueue<Integer> pq = new PriorityQueue<>();

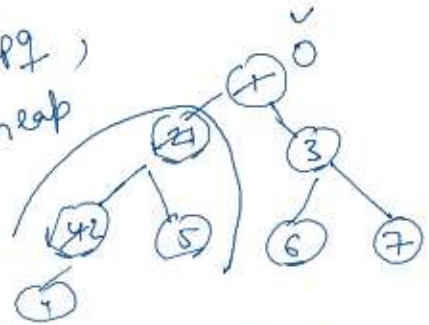
collections.reverseOrder();

[[minheapify()]]

1 2 3 4 5 6 7

O(logn)

min pq ,
minheap



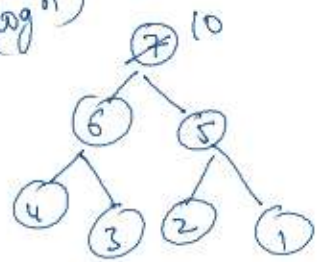
pq.peek() = 1

pq.poll() = 1
.remove.

pq.add(0)

[maxheapify()]

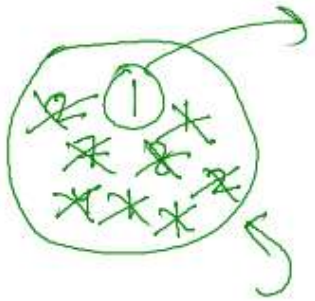
O(logn)



10

Qus Stone game

[2 7 4 1 8 1]
0 1 2 3 4 5

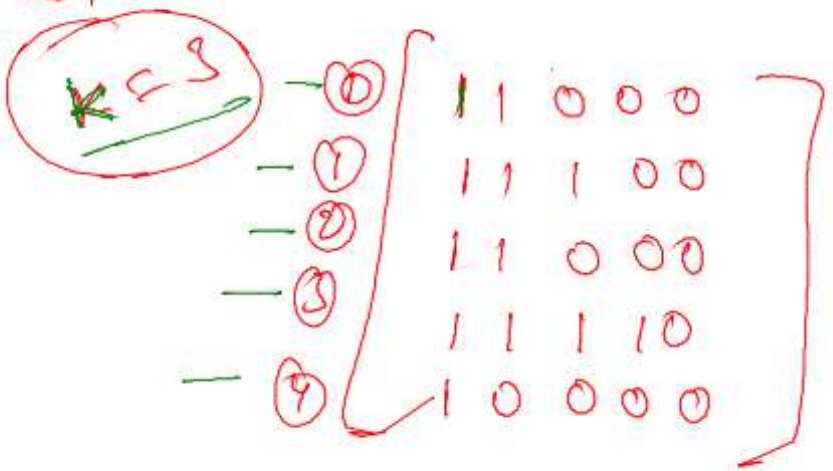


[2 7 4 1 8 1 1]

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for(int i=0;i<n;i++){
        arr[i]= scn.nextInt();
    }

    PriorityQueue<Integer> maxPQ = new PriorityQueue(Collections.reverseOrder());

    for(int num : arr){
        maxPQ.add(num);
    }

    while(maxPQ.size()>1){
        int a = maxPQ.poll();
        int b = maxPQ.poll();

        if(a!=b){
            int val = Math.abs(a-b);
            maxPQ.add(val);
        }
    }

    System.out.println(maxPQ.size()>0?maxPQ.peek():"0");
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class shc
}
```

a = 8 4 2 1 1
b = 8 1 1 1

Q. weak soldiers



0's = civilians
1's = soldiers

[4 2 3]