Experiment1.3

Student Name: Sameer Anand **UID:20BCS9834**
Branch: BE CSE **Section/Group: 20BCS_DM_714**
Semester: 6TH
Subject Name: CC LAB **Subject Code: 20CSP-351**

1. **Aim: To implement the concept of Heap model**

Leetcode code and output:

a. Kth Largest Element in an Array

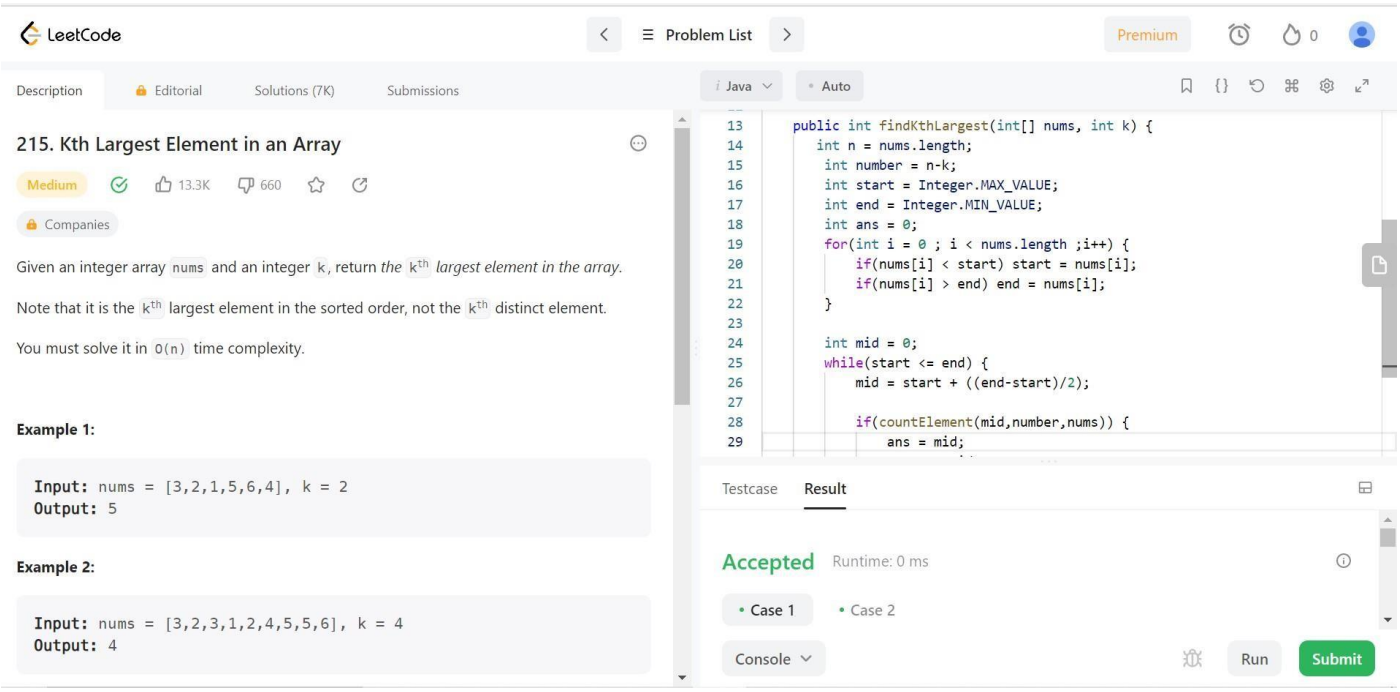
```
class Solution {    public static boolean countElement(int mid , int
number , int nums[]) {         int count = 0;         for(int i = 0; i <
nums.length ; i++) {             if(mid > nums[i]) {
count++;
            }         }         if(count
<= number) return true;         else
return false;
    }          public int findKthLargest(int[]
nums, int k) {        int n = nums.length;
int number = n-k;
        int start = Integer.MAX_VALUE;
int end = Integer.MIN_VALUE;             int
ans = 0;
        for(int i = 0 ; i < nums.length ;i++) {
if(nums[i] < start) start = nums[i];
if(nums[i] > end) end = nums[i];
        }
int mid = 0;
while(start <= end) {
            mid = start + ((end-start)/2);

if(countElement(mid,number,nums)) {
ans = mid;                 start = mid+1;

}else {
                end = mid - 1;
            }
}
```

```
        return ans;
    }
}
```

OUTPUT:



b. Last Stone Weight

Code:

```
class Solution {
    public int lastStoneWeight(int[] stones) {
          PriorityQueue <Integer> maxheap=new PriorityQueue<>(Collections.reverseOrder());
for (int i = 0; i < stones.length ; i++) {
            maxheap.add(stones[i]);
        }       while
(maxheap.size()!=1){
int x=maxheap.poll();
int y=maxheap.poll();
if(x!=y){
              maxheap.add(x-y);
```

```
        }          else
if(x==y){
maxheap.add(0);
        }
    }              return
maxheap.poll();
   } }
```

OUTPUT: