DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Experiment-1.3

Student Name: Kanishk Soni **UID: 20BCS9398**
Branch: BE-CSE **Section/Group:20BCS_DM-708B**
Semester: 6th **Subject Name: Competitive Coding-II**
Subject Code: 20CSP-351

AIM: To demonstrate the concept of Heap model

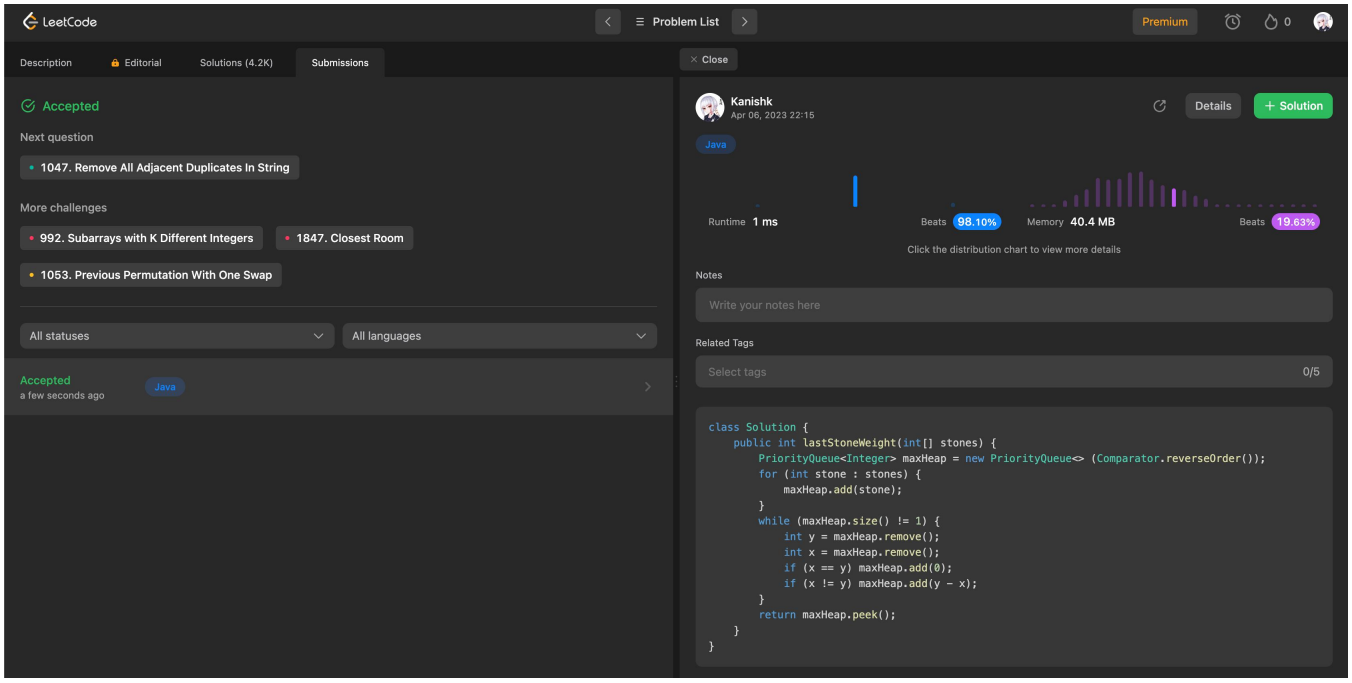
Problem1: Last stone weight
https://leetcode.com/problems/last-stone-weight/

Program Code:

```
class Solution {
    public int lastStoneWeight(int[] stones) {
        PriorityQueue<Integer> maxHeap = new PriorityQueue<>
(Comparator.reverseOrder());
        for (int stone : stones) {
            maxHeap.add(stone);
        }
        while (maxHeap.size() != 1) {
            int y = maxHeap.remove();
            int x = maxHeap.remove();
            if (x == y) maxHeap.add(0);
            if (x != y) maxHeap.add(y – x);
        }
        return maxHeap.peek();
    }
```

}

Output:



Probem2: Cheapest flights with k shops
https://leetcode.com/problems/cheapest-flights-within-k-stops/

Program Code:

```
class Solution {
    public int findCheapestPrice(int n, int[][] flights, int src, int dst, int k) {
        int [] prev = new int[n];
        Arrays.fill(prev,Integer.MAX_VALUE);
        prev[src]=0;
        for(int i=0;i<=k;i++){
            int cur[]= new int[n];
            for(int j=0;j<n;j++){
```

```
            cur[j]= prev[j];
        }
        for(int e[]:flights){
            int u=e[0],v=e[1],wt=e[2];
            if(prev[u] != Integer.MAX_VALUE && prev[u]+wt <cur[v]){
                cur[v] = prev[u]+wt;
            }
        }
        for(int j=0;j<n;j++){
            prev[j]=cur[j];
        }
    }
        return prev[dst]==Integer.MAX_VALUE ?-1:prev[dst];
    }
}
```

Output: