Experiment1.3

Student Name: Himanshu **UID:** 20BCS7944
Branch: BE-CSE **Section/Group:** 905/A
Semester: 6th **Date of Performance:** 22/02/2023
Subject Name: Competitive Coding-II **Subject Code:** 20CSP-351

1. Aim:

To implement the concept of string heap model.

2. Objective:

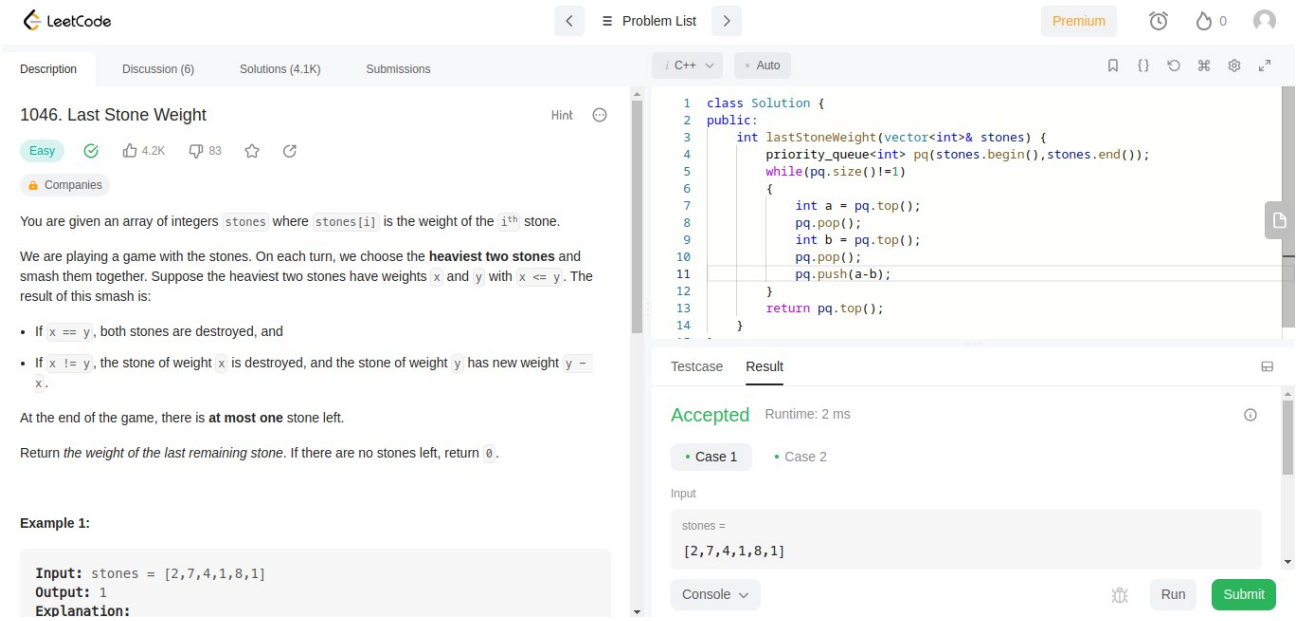
- The objective is to build problem solving capability and to learn the basic concepts of data structures.
- The implementation of last weight stone which shows and brushes up the concept of heap data structures.
- The implementation of cheapest flight within k-stops + in which the concept of kmp was introduced.

3. LeetCode code and output:

- **LAST STONE WEIGHT-**

```
class Solution {
public:
	int lastStoneWeight(vector<int>& stones) {
	priority_queue<int> pq(stones.begin(),stones.end());
		while(pq.size()!=1){
		int a = pq.top();
		pq.pop();
		int b = pq.top();
		pq.pop();
		pq.push(a-b);
	}
		return pq.top();
	}
};
```

OUTPUT:



- **CHEAPEST FLIGT WITHIN K-STOPS**

```
class Solution {
const int inf = 1e8;
int dp[101][101];

public:
int dfs(int node, int k, int dest, vector<vector<int>> &cost, vector<int> adj[])
{
if (k < 0)
return inf;

if (node == dest)
return 0;

if (dp[node][k] != -1)
return dp[node][k];

int ans = inf;
for (auto i : adj[node])
ans = min(ans, cost[node][i] + dfs(i, k - 1, dest, cost, adj));
```

```
return dp[node][k] = ans;
}
int findCheapestPrice(int n, vector<vector<int>> &flights, int src, int dst, int k){
memset(dp, -1, sizeof dp);

vector<vector<int>> cost(n, vector<int>(n));
vector<int> adj[n];
for (auto e : flights){
adj[e[0]].push_back(e[1]);
cost[e[0]][e[1]] = e[2];
}

int ans = dfs(src, k + 1, dst, cost, adj);
return ans == inf ? -1 : ans; }
};
```

OUTPUT: