Experiment 6

Student Name: Sweta Sharma **UID: 22BCS11536**
Branch: BE-CSE **Section/Group: 22BCS-IOT-637(B)**
Semester: 6th **Date of Performance: 27/02/25**
Subject Name: ADVANCED PROGRAMMING LAB - 2 **Subject Code: 22CSP-351**

PROGRAM-1

1) **Aim:** Is Graph Bipartite.

2) **Implementation/Code:**

```
class Solution {
    bool check(int node, vector<vector<int>> &adj, vector<int> &color){
        int m = adj.size();
        queue<int> q;
        q.push(node);
        color[node] = 0;

        while(!q.empty()){
            int x = q.front();
            q.pop();
            for(int adjNode: adj[x]){
                if(color[adjNode] == -1){
                    color[adjNode] = 1 - color[x];
                    q.push(adjNode);
                }
                else if(color[adjNode] == color[x]) return false;
            }
        }
        return true;
    }
public:
    bool isBipartite(vector<vector<int>>& graph) {
        int n = graph.size();
        vector<int> color(n, -1);
```

```
            for(int i=0; i<n; i++){
                if(color[i] == -1){
                    if(!check(i, graph, color)) return false;
                }
            }
            return true;
        }
};
```

3) **Output:**

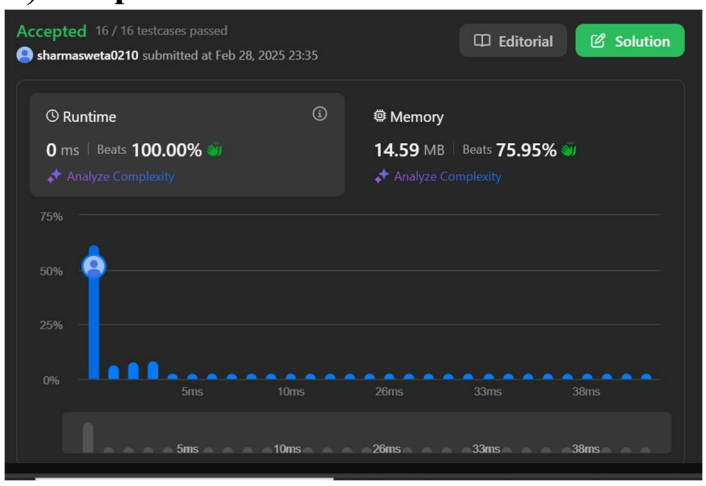
PROGRAM-2

1) **Aim:**Gray Code.

2) **Implementation/Code:**

```
class Solution {
public:
    vector<int> grayCode(int n) {
        vector<int> result;
        for (int i = 0; i < (1 << n); ++i) {
            result.push_back(i ^ (i >> 1));
        }
        return result;
    }
};
```

3) **Output:**



PROGRAM-3

1) **Aim:** Group the People Given the Group Size They Belong To.

2) **Implementation/Code** :class Solution {

```
public:
    vector<vector<int>> groupThePeople(vector<int>& groupSizes) {
        int n=groupSizes.size();
        vector<vector<int> > ans;
        unordered_map<int,vector<int>> mp ;
        for (int i=0;i<n;i++){
            mp[groupSizes[i]].push_back(i);

        }
        for (auto x: mp){
            int groupsize=x.first;
            vector<int> a=x.second;
            if (a.size()==groupsize){
                ans.push_back(a);
            }
            else{
                int c=groupsize;
                int n=a.size();
```

```
            int i=0;
            while (i!=n){
               vector<int> b;
             for (int j =i;j<i+c;j++){
               b.push_back(a[j]);
             }
             i=i+c;
             ans.push_back(b);

            }
          }

      }
      return ans;

   }
};
```

3) **Output:**