Experiment 6

Student Name: Anshul Verma **UID: 22BCS11915**
Branch: CSE **Section/Group: 637-B**
Semester: 6th **Date of Performance:27/2/25**
Subject Name: Advanced Programming - 2 **Subject Code: 22CSH-351**

Ques 1: Aim:

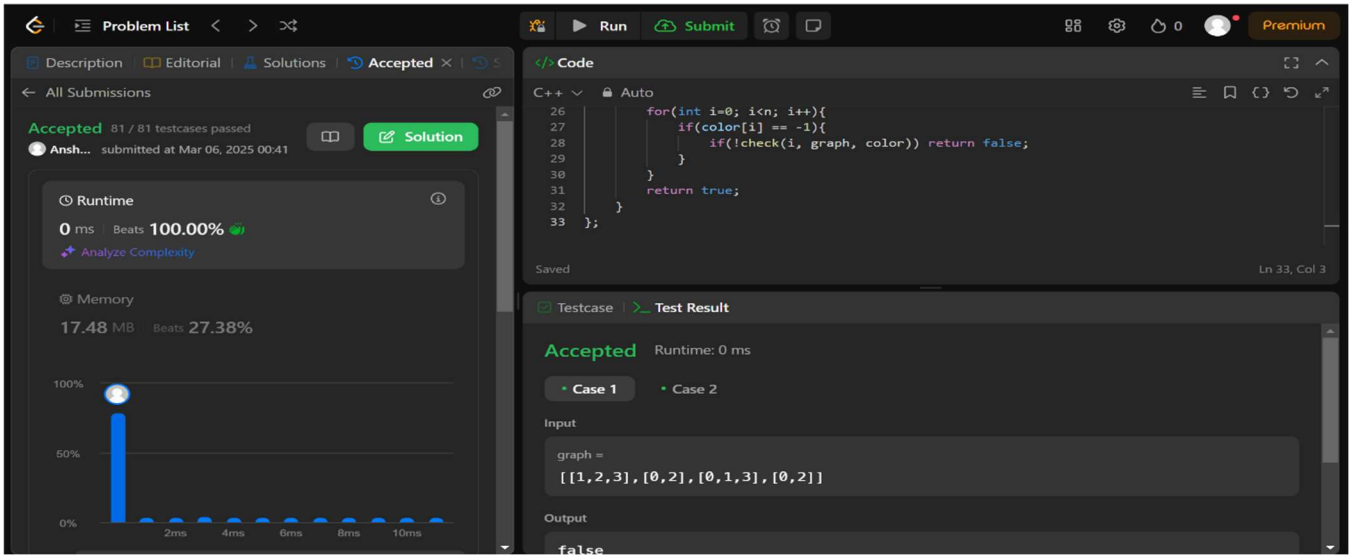
is graph bipartite?

Code:

```
#include <vector>
#include <queue>
using namespace std;

class Solution {
public:
    bool isBipartite(vector<vector<int>>& graph) {
        int n = graph.size();
        vector<int> color(n, -1); // -1 means uncolored, 0 and 1 are the two colors

        for (int i = 0; i < n; i++) {
            if (color[i] == -1) { // If the node is not colored, perform BFS
                queue<int> q;
                q.push(i);
                color[i] = 0; // Start coloring with 0

                while (!q.empty()) {
                    int node = q.front();
                    q.pop();

                    for (int neighbor : graph[node]) {
                        if (color[neighbor] == -1) {
                            color[neighbor] = 1 - color[node]; // Assign the opposite color
```

```
                    q.push(neighbor);
                } else if (color[neighbor] == color[node]) {
                    return false; // If two adjacent nodes have the same color, the graph is not
bipartite
                }
            }
        }
    }
    return true;
  }
};
```

Submission Screenshot:



Link: https://leetcode.com/problems/is-graph-bipartite/submissions/1564121337/

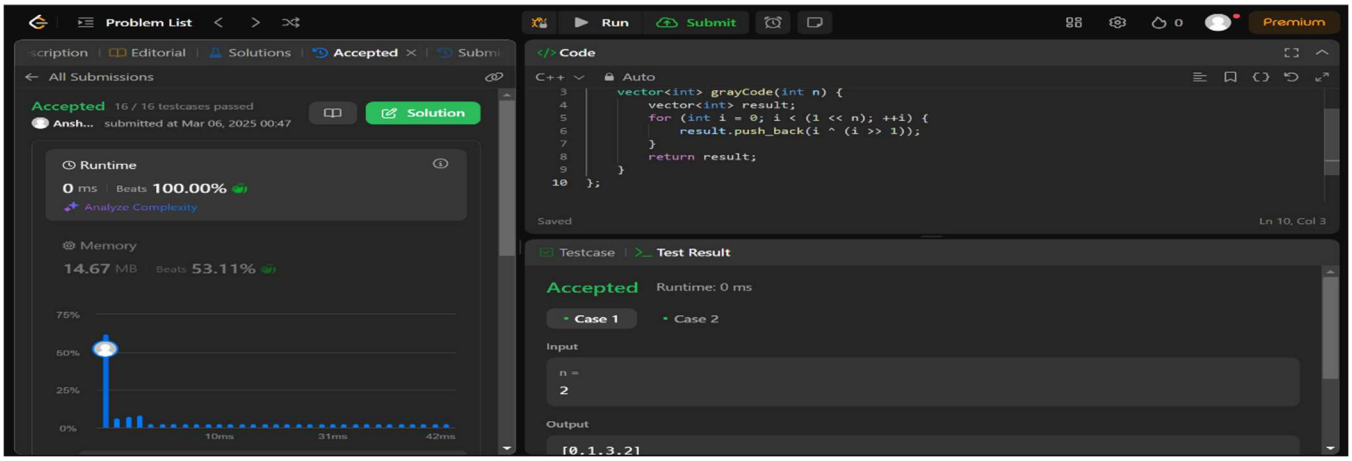
Ques 2:

Aim: Gray code

Code:

```
#include <vector>
using namespace std;

class Solution {
public:
    vector<int> grayCode(int n) {
        vector<int> result;
        int total = 1 << n; // 2^n
        for (int i = 0; i < total; i++) {
            result.push_back(i ^ (i >> 1)); // Generate Gray code
        }
        return result;
    }
};
```

Submission Screenshot:



Link: https://leetcode.com/problems/gray-code/submissions/1564126208/

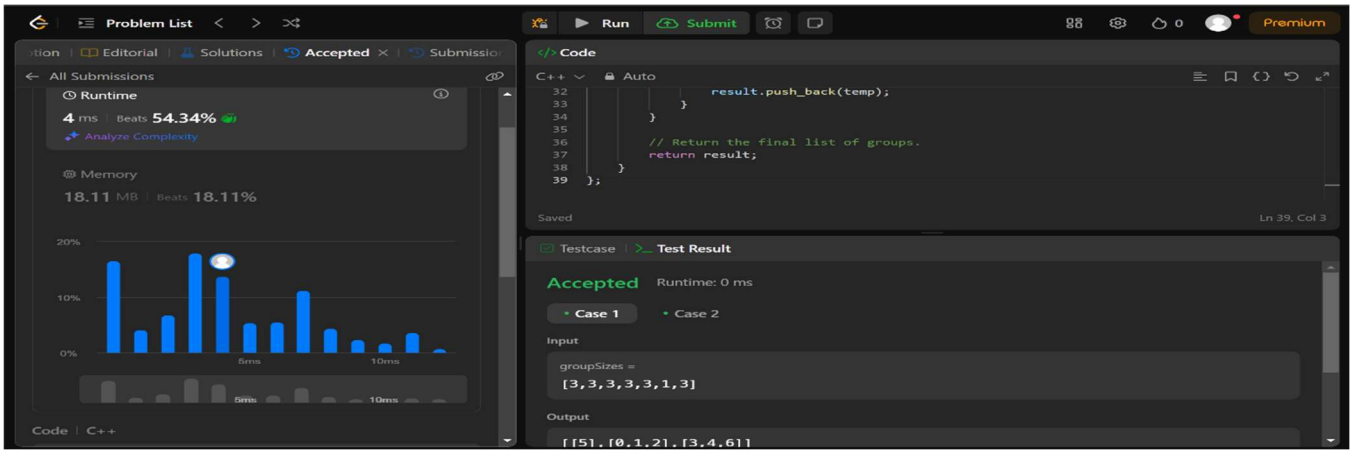
Ques 3:

Aim: Group the People Given the Group Size They Belong To

Code:

```
#include <vector>
#include <unordered_map>
using namespace std;

class Solution {
public:
    vector<vector<int>> groupThePeople(vector<int>& groupSizes) {
        unordered_map<int, vector<int>> groups;
        vector<vector<int>> result;

        for (int i = 0; i < groupSizes.size(); i++) {
            groups[groupSizes[i]].push_back(i);
            if (groups[groupSizes[i]].size() == groupSizes[i]) {
                result.push_back(groups[groupSizes[i]]);
                groups[groupSizes[i]].clear();
            }
        }
        return result;
    }
};
```

Submission Screenshot:



Link: https://leetcode.com/problems/group-the-people-given-the-group-size-they-belong-to/submissions/1564128764/