```cpp
// Assignment 11 - Bipartite / Graph Coloring
// Greedy Algorithm

class Solution {
public:
    void DFS(int garden, vector<int>& answer, vector<vector<int>>& adj_list) {
        // In each garden, plant one of 4 types of flowers
        vector<bool> typeFlowers(5, false);

        for (int neighbor : adj_list[garden]) {
            if (answer[neighbor] != 0) {
                typeFlowers[answer[neighbor]] = true;
            }
        }

        for (int flowerType = 1; flowerType <= 4; flowerType++) {
            if (!typeFlowers[flowerType]) {
                answer[garden] = flowerType;
                break;
            }
        }
    }

    vector<int> gardenNoAdj(int n, vector<vector<int>>& paths) {
        // Return an array answer, where answer[i] is the type of flower
        // planted in the (i+1)th garden
        vector<int> answer(n, 0);

        // Adjacency list
        vector<vector<int>> adj_list(n);
        for (const auto& path : paths) {
            int x = path[0] - 1;
            int y = path[1] - 1;
            adj_list[x].push_back(y);
            adj_list[y].push_back(x);
        }

        // Iterate over each garden
        for (int garden = 0; garden < n; garden++) {
            if (answer[garden] == 0) {
                DFS(garden, answer, adj_list);
            }
        }

        return answer;
    }
};
```