Design and Analysis of Algorithms

Name: Harsh Brahmecha
PRN: 20220802003

LAB 5

To write a code to solve graph tree colouring problem using back- tracking method.

CODE :

```c
#include <stdio.h>

// Define the number of vertices in the graph
#define V 4

// Function to check if it's safe to color the vertex 'v' with color 'c'
int isSafe(int v, int graph[V][V], int color[], int c) {
    for (int i = 0; i < V; i++) {
        if (graph[v][i] && c == color[i]) {
            return 0;
        }
    }
    return 1;
}

// Function to solve the graph coloring problem using backtracking
int graphColoringUtil(int graph[V][V], int m, int color[], int v) {
    if (v == V) {
        return 1; // All vertices are colored
    }

    for (int c = 1; c <= m; c++) {
        if (isSafe(v, graph, color, c)) {
            color[v] = c;

            if (graphColoringUtil(graph, m, color, v + 1)) {
```

```
                return 1; // If the next vertex can be colored, we are done
            }

            color[v] = 0; // Backtrack
        }
    }

    return 0; // If no color can be assigned to this vertex
}

// Function to solve the graph coloring problem
int graphColoring(int graph[V][V], int m) {
    int color[V];
    for (int i = 0; i < V; i++) {
        color[i] = 0; // Initialize colors as 0 (not assigned)
    }

    if (graphColoringUtil(graph, m, color, 0) == 0) {
        printf("Solution does not exist with %d colors\n", m);
        return 0;
    }

    printf("Solution exists with the following colors:\n");
    for (int i = 0; i < V; i++) {
        printf("Vertex %d --> Color %d\n", i, color[i]);
    }
```

```
int main() {
    int graph[V][V] = {
        {0, 1, 1, 1},
        {1, 0, 1, 0},
        {1, 1, 0, 1},
        {1, 0, 1, 0}
    };
    int m = 3; // Number of colors

    graphColoring(graph, m);
    return 0;
}
```

OUTPUT :

```
Solution exists with the following colors:
Vertex 0 --> Color 1
Vertex 1 --> Color 2
Vertex 2 --> Color 3
Vertex 3 --> Color 2
```