Practical 8 (DAA)

Name : Mohammad Ejaz Mehendi

Section : A3-B3

Roll no : 48

Aim: Implement Graph Colouring algorithm use Graph colouring concept.

Problem Statement:

A GSM is a cellular network with its entire geographical range divided into hexadecimal cells. Each cell has a communication tower which connects with mobile phones within cell. Assume this GSM network operates in different frequency ranges. Allot frequencies to each cell such that no adjacent cells have same frequency range.

Consider an undirected graph G = (V, E) shown in fig. Find the colour assigned to each node using Backtracking method. Input is the adjacency matrix of a graph G(V, E), where V is the number of Vertices and E is the number of edges.

Code :

#include <stdio.h>

#include <stdbool.h>

#define V 6

void printSolution(int color[]) {

printf("Solution Exists: Following are the assigned colors:\n");

for (int i = 0; i < V; i++) {

printf(" Vertex %d ---> Color %d\n", i, color[i]);

}

printf("\n");

}

bool 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 false;

}

}

return true;

}

bool graphColoringUtil(int graph[V][V], int m, int color[], int v) {

if (v == V) {

return true;

}

for (int c = 1; c <= m; c++) {

if (isSafe(v, graph, color, c)) {

color[v] = c;

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

return true;

}

color[v] = 0;

}

}

return false;

}

bool graphColoring(int graph[V][V], int m) {

int color[V];

for (int i = 0; i < V; i++) {

color[i] = 0;

}

if (graphColoringUtil(graph, m, color, 0) == false) {

printf("Solution does not exist for %d colors.\n", m);

return false;

}

printSolution(color);

return true;

}

int main() {

int graph[V][V] = {

{0, 1, 1, 0, 0, 0},

{1, 0, 0, 1, 0, 0},

{1, 0, 0, 1, 1, 0},

{0, 1, 1, 0, 0, 1},

{0, 0, 1, 0, 0, 1},

{0, 0, 0, 1, 1, 0}

};

int m = 3;

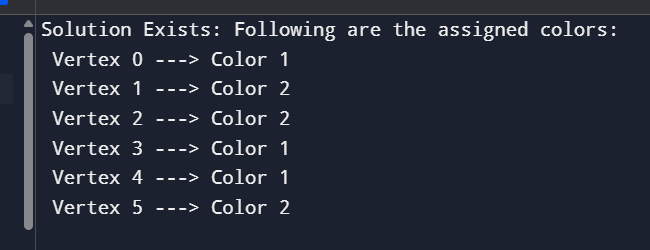
graphColoring(graph, m);

return 0;

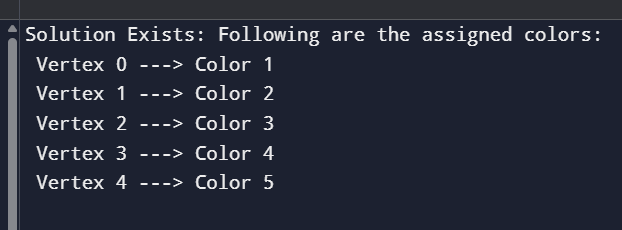
}

Output :

For Graph 1



For Graph 2



Screenshot :

