|  |
| --- |
| **NAME:** GOKULRAJ V |
| **ROLL\_NO:** 1905014 |

|  |
| --- |
| **DATE:** 09/09/2022 |
| **EXP\_NO:** 04 |

|  |
| --- |
| **CONSTRAINT SATISFACTION PROBLEM** |

**Aim**:

To solve the Map Coloring problem with the help of A\* Algorithm.

**Problem Statement:**

Consider the graph with 8 nodes A1, A2, A3, A4, H, T, F1, F2. A i is connected to A i+1 for all i, each Ai is connected to H, H is connected to T, and T is connected to each F i . Find a 3-coloring of this graph by using the following strategy: backtracking with conﬂict-directed back-jumping, the variable order A1, H, A4, F1, A2, F2, A3, T, and the value order R, G, B.

**CODE:**

#include<bits/stdc++.h>

using namespace std;

#define n 8

bool isSafe(bool graph[n][n], int color[])

{

for (int i = 0; i < n; i++)

for (int j = i + 1; j < n; j++)

if (graph[i][j] && color[j] == color[i])

return false;

return true;

}

void printSolution(int color[])

{

cout << "The assigned colors for the nodes are : \n nodes : A1 A2 A3 A4 H T F1 F2 \n color :";

char arr[3][1]={'R','G','B'};

for (int i = 0; i < n; i++)

cout << " " << arr[color[i]-1][0];

cout << "\n";

}

bool graphColoring(bool graph[n][n], int m, int i, int color[n]);

int main()

{

bool graph[n][n] = {

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

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

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

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

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

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

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

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

};

int m = 3; // Number of colors

int color[n];

for (int i = 0; i < n; i++)

color[i] = 0;

if (!graphColoring(graph, m, 0, color))

cout << "Solution does not exist";

return 0;

}

bool graphColoring(bool graph[n][n], int m, int i, int color[n])

{

if (i == n)

{

if(isSafe(graph, color))

{

printSolution(color);

return true;

}

//printSolution(color);

return false;

}

for (int j = 1; j <= m; j++)

{

color[i] = j;

if (graphColoring(graph, m, i + 1, color))

return true;

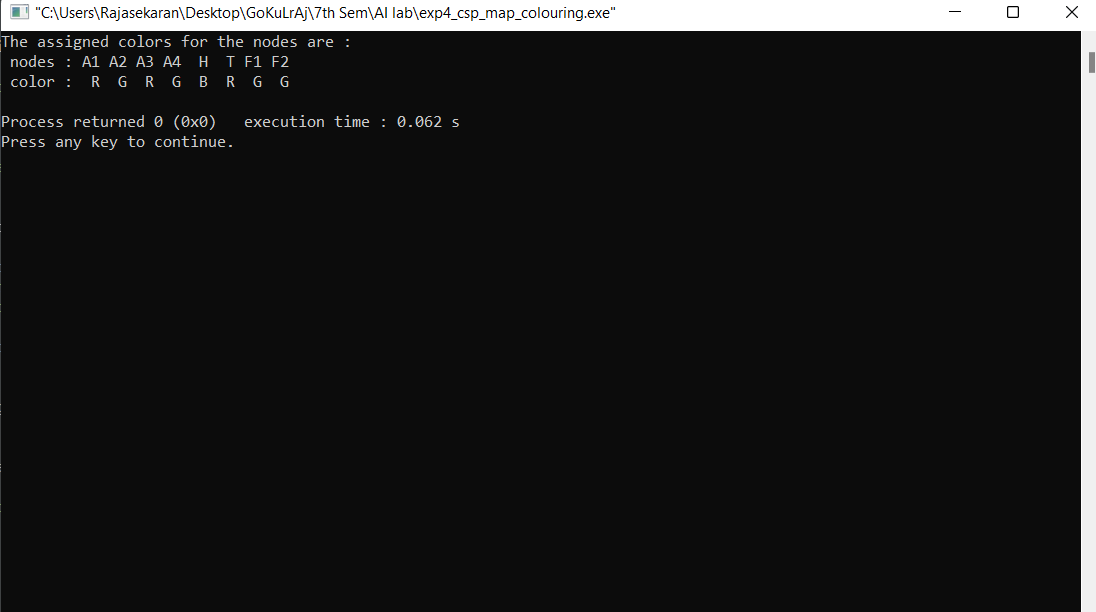
color[i] = 0;

}

return false;

}

**OUTPUT:**



**Result:**

Thus, the code has been successfully executed.