Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Summer, Year:2021), B.Sc. in CSE (Evn)

LAB REPORT NO 04

Course Title: Artificial Intelligence Lab

Course Code: CSE-404 Section: 183-EB

Lab Experiment Name: Implementation of Constraint satisfaction problem map coloring problem

Student Details

	Name	ID
1.	Md Safiullah	183015027
2.		
3.		

Lab Date : 07-07-20201

Submission Date : 14-07-2021

Course Teacher's Name : Md Atikuzzamn

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: …… **Signature:.....**

Comments:.....	**Date:.....**

Title: Constraint satisfaction problem map coloring problem

Theory: Constraint satisfaction problems require that all a problem's variables be assigned values, out of a finite domain, that result in the satisfying of all constraints. The map-coloring CSP, for example, is to assign a color to each region of a map such that any two regions sharing a border have different colors.

Objective: Implement this CSP map coloring problem using java

Procedure:

Constraint satisfaction map coloring problem is problem that need some variables whish is map, where hove some data which represent as vertex in a adjList

So then hove some color which are coloring or assign to vertex, But vertex need to different color

New setting up how to implement this algorithm.

Step 1: At first need to declare some variable such as graph, color, length, then declare a method like graphColor(). In this method initialize this variables and then use a try catch. When call solve() and passing 0 then if it return true then should print not possible, if is not then using another function which is display() for printing possible color on vertex

Step 2: In this step declare solve() which passing parameter as 0 if it is not equal to graph length of V then throw a new message and also checking in if condition which is a isPossible() method Where passing two parameters as v and c, v is vertex and c is a color. In this if condition assign the color[v] = c then call the solve() method and passing the updated v value then again color[v] assign to 0

Step 3: In this step now declare isPossible() method then using a for loop is I = 0 to graph length and in this loop just checking if graph[v][i] equal to 1 and also passed color c equal to color[i] then return false. If it not return false then should be return true which is the indicating possible color.

Step 4: In this step declare display method and printing all possible vertex of color.

Step 5: In this step we need to declare our main method which is graphColor in main function using Class object also passing the graph value by using user input and color.

Step 6: End

Now showing my code,

```
package csp.map.coloring;

import java.util.Scanner;

public class CSPMapColoring
{
    private int V, numOfColors;
    private int[] color;
    private int[][] graph;

    public void graphColor(int[][] g, int col)
    {
        V = g.length;
        numOfColors = col;
        color = new int[V];
        graph = g;

        try
        {
            solve(0);
            System.out.println("No solution");
        }
        catch (Exception e)
        {
            System.out.println(e.getMessage());
            display();
        }
    }

    public void solve(int v) throws Exception
    {
        if (v == V)
            throw new Exception("Solution found");
        for (int c = 1; c <= numOfColors; c++)
        {
            if (isPossible(v, c))
            {
                color[v] = c;
                solve(v + 1);
                color[v] = 0;
            }
        }
    }

```

```
46      public boolean isPossible(int v, int c)
47      {
48          for (int i = 0; i < V; i++)
49              if (graph[v][i] == 1 && c == color[i])
50                  return false;
51          return true;
52      }
53
54      public void display()
55      {
56          System.out.print("\nColors : ");
57          for (int i = 0; i < V; i++)
58              System.out.print(color[i] +" ");
59          System.out.println();
60      }
61
62      public static void main (String[] args)
63      {
64          Scanner scan = new Scanner(System.in);
65          CSPMapColoring gc = new CSPMapColoring();
66
67          System.out.println("Enter length:\n");
68          int V = scan.nextInt();
69
70          System.out.println("\nEnter matrix\n");
71          int[][] graph = new int[V][V];
72          for (int i = 0; i < V; i++)
73              for (int j = 0; j < V; j++)
74                  graph[i][j] = scan.nextInt();
75
76          System.out.println("\nEnter number of colors:");
77          int c = scan.nextInt();
78
79          gc.graphColor(graph, c);
80
81      }
82  }
83
```

Output:

```
run:
Enter length:

4

Enter matrix

0 1 1 1
1 0 1 0
1 1 0 1
1 0 1 0

Enter number of colors:
3
Solution found

Colors : 1 2 3 2
BUILD SUCCESSFUL (total time: 9 seconds)
```

Conclusion: this is vary simple to implement. I learn this algorithm also learn how to implement.