**GRAPH THEORY AND IT’S APPLICATIONS**

**ASSIGNMENT – 2**

NAME : Sanmat Sanjayakumar Payagoudar

SRN : PES1UG20CS385

**PROBLEM STATEMENT :**

1. Consider the map of a country. Use a graph coloring algorithm to find the minimum number of colors required to color the states and show one such coloring.

**ALGORITHM EXPLANATION :**

  A utility function is\_Safe() is written to check if the specified color assigned is safe for the given vertex v.

  The graph\_Colour() is a user defined recursive function to solve coloring problems

**CODE :**

class graphColor():

    def \_\_init\_\_(s, ver):

        s.V = ver

        s.graph = [[0 for column in range(ver)] for row in range(ver)]

    def isSafe(s, v, clr, c):

        for i in range(s.V):

            if s.graph[v][i] == 1 and clr[i] == c:

                return False

        return True

    def graphColour(s, m, clr, v):

        if v == s.V:

            return True

        for c in range(1, m + 1):

            if s.isSafe(v, clr, c) == True:

                clr[v] = c

                if s.graphColour(m, clr, v + 1) == True:

                    return True

                clr[v] = 0

    def graphColouring(s, m):

        clr = [0] \* s.V

        if s.graphColour(m, clr, 0) == None:

            return False

        print("Solution exists! Colors assigned are:")

        for c in clr:

            print(c, end=' ')

        return True

if \_\_name\_\_ == '\_\_main\_\_':

    g = graphColor(5)

    g.graph = [[0, 1, 1, 1, 1], [1, 0, 0, 1, 0], [1, 0, 0, 0, 0], [1, 1, 0, 0, 0], [1, 0, 0, 0, 0]]

    m = 3

    g.graphColouring(m)