CSE\_A.I\_Lab\_CLP4\_Shahidul-213902017

**py.Code**

class GraphColoring:

    def \_\_init\_\_(self):

        self.numOfColors = 0

        self.color = {}

        self.graph = {}

    def graphColor(self, g, noc):

        self.numOfColors = noc

        self.color = {}

        self.graph = g

        try:

            self.solve('westernAustralia')

            print("No solution")

        except:

            print("\nSolution exists ")

            self.display()

    def solve(self, v):

        if len(self.color) == len(self.graph):

            raise Exception("Solution found")

        for c in range(1, self.numOfColors + 1):

            if self.isPossible(v, c):

                self.color[v] = c

                self.solve(self.getNextVertex())

                del self.color[v]

    def isPossible(self, v, c):

        for neighbor in self.graph[v]:

            if neighbor in self.color and self.color[neighbor] == c:

                return False

        return True

    def getNextVertex(self):

        for vertex in self.graph:

            if vertex not in self.color:

                return vertex

    def display(self):

        textColor = ["", "RED", "GREEN", "BLUE", "YELLOW", "ORANGE", "PINK", "BLACK", "BROWN", "WHITE", "PURPLE", "VIOLET"]

        print("\nAreas and Colors:")

        for area, c in self.color.items():

            print(area + ":", textColor[c])

    @staticmethod

    def main():

        print("Graph Coloring Algorithm Test\n")

        gc = GraphColoring()

        graph = {

            'westernAustralia': ['NorthernTerritory', 'SouthAustralia'],

            'NorthernTerritory': ['westernAustralia', 'SouthAustralia', 'Queensland'],

            'SouthAustralia': ['westernAustralia', 'NorthernTerritory', 'Queensland', 'NewSouthWales', 'Victoria'],

            'Queensland': ['NorthernTerritory', 'SouthAustralia', 'NewSouthWales'],

            'NewSouthWales': ['SouthAustralia', 'Queensland', 'Victoria'],

            'Victoria': ['SouthAustralia', 'NewSouthWales', 'Tasmania'],

            'Tasmania': ['Victoria']

        }

        c = 4  # Number of colors

        gc.graphColor(graph, c)

if \_\_name\_\_ == "\_\_main\_\_":

    GraphColoring.main()

**Output**

Graph Coloring Algorithm Test

Solution exists

Areas and Colors:

westernAustralia: RED

NorthernTerritory: GREEN

SouthAustralia: BLUE

Queensland: RED

NewSouthWales: GREEN

Victoria: RED

Tasmania: GREEN