Priyanka Srinivas | RA1911026010014

Artificial Intelligence Lab

Exercise 2: The Graph Colouring Problem

AIM: To solve the graph colouring problem

INTRODUCTION:

The Graph Colouring problem is to determine if an undirected graph can be coloured with at most m colours while no two neighbouring vertices of the graph are coloured with the same colour, given an undirected graph and an integer m. The assignment of colours to all vertices in a graph is referred to as colouring.

ALGORITHM:

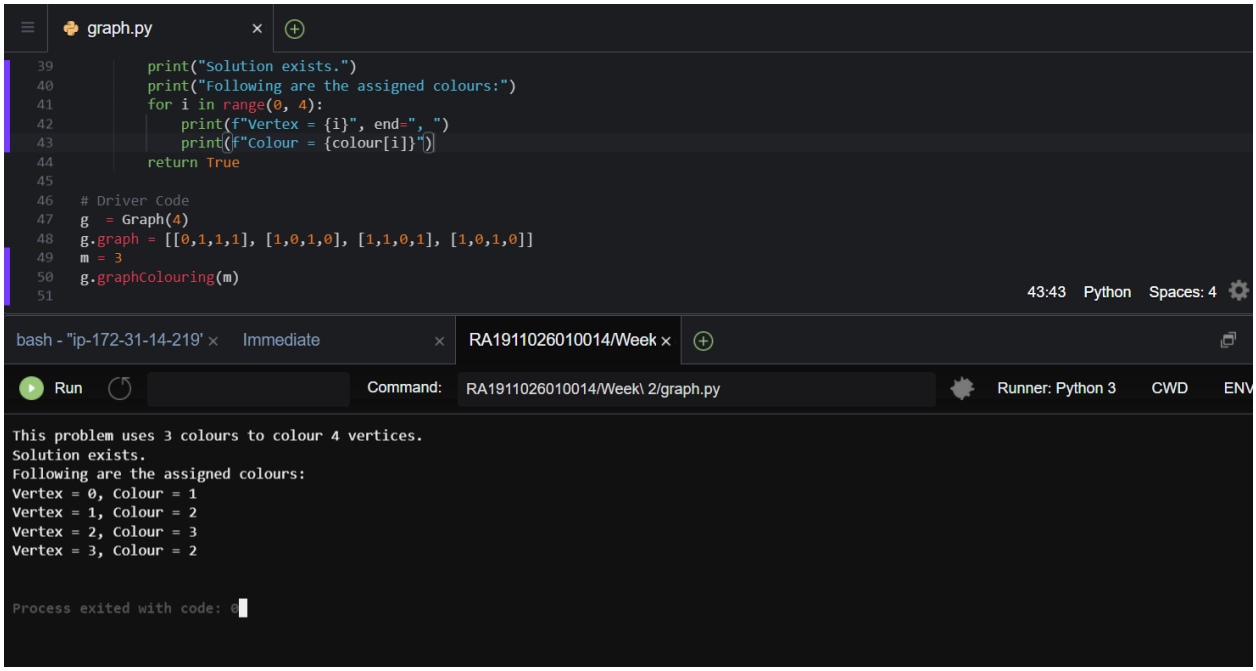
1. Make a recursive function that accepts the graph, the current index, the number of vertices, and the output colour array as inputs.
2. If the number of vertices equals the current index. In the output array, print the colour configuration.
3. A colour can be assigned to a vertex (1 to m).
4. Check if the configuration is safe for each provided colour (i.e. check if nearby vertices do not have the same colour), then recursively run the function with the next index and number of vertices.
5. Break the loop and return true if any recursive function returns true.
6. Return false if no recursive function returns true.

PROGRAM:

```
class Graph():
    def __init__(self, vertices):
        self.V = vertices
        self.graph = [[0 for column in range(vertices)]
                              for row in range(vertices)]

    # A utility function to check if the current colour assignment
    # is valid for vertex v
    def isValid(self, v, colour, c):
        for i in range(self.V):
            if self.graph[v][i] == 1 and colour[i] == c:
                return False
```

```
        return True

    # A recursive utility function to solve m
    # colouring  problem
    def graphColourUtil(self, m, colour, v):
        if v == self.V:
            return True

        for c in range(1, m+1):
            if self.isValid(v, colour, c):
                colour[v] = c
                if self.graphColourUtil(m, colour, v+1):
                    return True
                colour[v] = 0

    def graphColouring(self, m):
        colour = [0] * self.V
        if self.graphColourUtil(m, colour, 0) == False:
            return False

        # Print the solution
        print(f"This problem uses {m} colours to colour {self.V}
vertices.")
        print("Solution exists.")
        print("Following are the assigned colours:")
        for i in range(0, 4):
            print(f"Vertex = {i}", end=", ")
            print(f"Colour = {colour[i]}")
        return True

# Driver Code
g = Graph(4)
g.graph([0,1,1,1], [1,0,1,0], [1,1,0,1], [1,0,1,0]
m = 3
g.graphColouring(m)
```

OUTPUT:



RESULT:

The graph colouring problem was implemented successfully.