RA1911033010075

VAIBHAV MISHRA

# EXPERIMENT - 2

AIM

To develop agent programs for real world problems

## M COLORING PROBLEM

ALGORITHM

1) Create a recursive function that takes current index, number of vertices and output color array.

2) If the current index is equal to number of vertices. check if the output color configuration is safe. i.e, check if the adjacent do not have same color. If the condition are met, print the configuration and break.

3) Assign a color to a vertex (1 to m)

4) For every assigned color recursively call the function with next index and number of vertices.

5) If any recursive function returns true break the loop and return true.

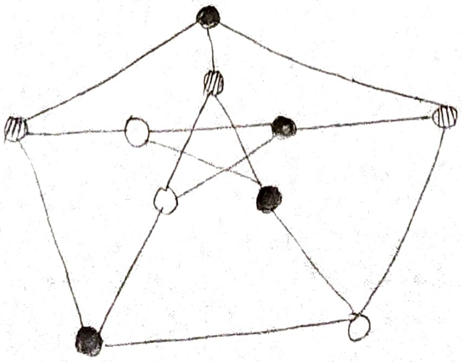
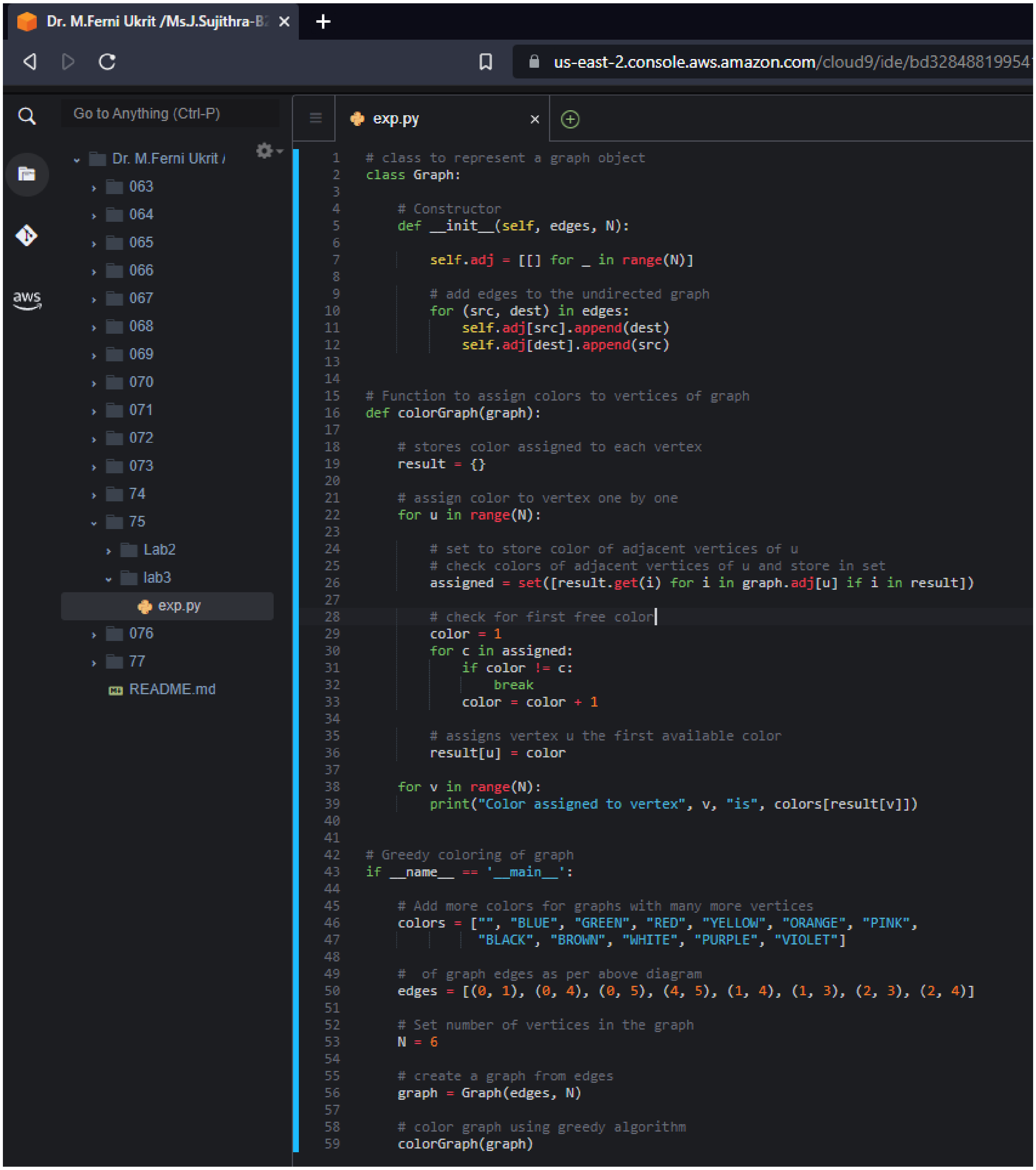


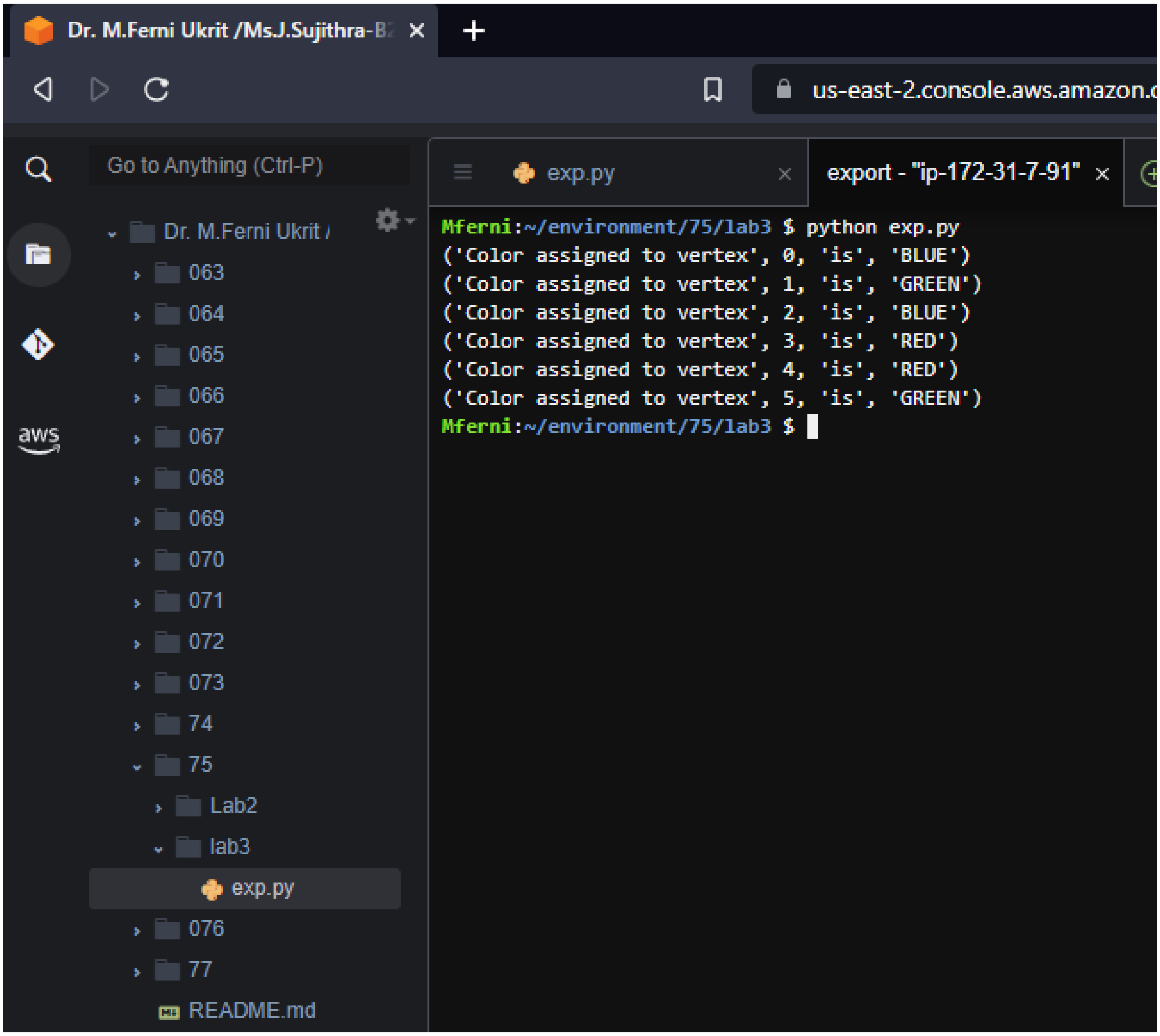
Fig. Example of graph that can be coloured with three different colours.

## PROGRAM :



```python
# class to represent a graph object
class Graph:

    # Constructor
    def __init__(self, edges, N):

        self.adj = [[] for _ in range(N)]

        # add edges to the undirected graph
        for (src, dest) in edges:
            self.adj[src].append(dest)
            self.adj[dest].append(src)


# Function to assign colors to vertices of graph
def colorGraph(graph):

    # stores color assigned to each vertex
    result = {}

    # assign color to vertex one by one
    for u in range(N):

        # set to store color of adjacent vertices of u
        # check colors of adjacent vertices of u and store in set
        assigned = set([result.get(i) for i in graph.adj[u] if i in result])

        # check for first free color
        color = 1
        for c in assigned:
            if color != c:
                break
            color = color + 1

        # assigns vertex u the first available color
        result[u] = color

    for v in range(N):
        print("Color assigned to vertex", v, "is", colors[result[v]])


# Greedy coloring of graph
if __name__ == '__main__':

    # Add more colors for graphs with many more vertices
    colors = ["", "BLUE", "GREEN", "RED", "YELLOW", "ORANGE", "PINK",
              "BLACK", "BROWN", "WHITE", "PURPLE", "VIOLET"]

    #  of graph edges as per above diagram
    edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]

    # Set number of vertices in the graph
    N = 6

    # create a graph from edges
    graph = Graph(edges, N)

    # color graph using greedy algorithm
    colorGraph(graph)
```

**OUTPUT :**



```
Mferni:~/environment/75/lab3 $ python exp.py
('Color assigned to vertex', 0, 'is', 'BLUE')
('Color assigned to vertex', 1, 'is', 'GREEN')
('Color assigned to vertex', 2, 'is', 'BLUE')
('Color assigned to vertex', 3, 'is', 'RED')
('Color assigned to vertex', 4, 'is', 'RED')
('Color assigned to vertex', 5, 'is', 'GREEN')
Mferni:~/environment/75/lab3 $
```