

# Artificial Intelligence

## Week 2

**AIM:**To implement Graph coloring problem.

**Problem statement :**

Graph-coloring problem:

Given a graph and colours we have to colour the graph using the given colours such that the constraints are satisfied.

**ALGORITHM:**

- 1) Create the graph with given edges and nodes.Store it in adjacency list.
- 2) We will be iterating through all the nodes to assign the required colors.
- 3) Color the first node with the first color.
- 4) From the next node, we will be checking with its neighboring nodes for each color and if the color is valid we store it into the dictionary with node as it's key and color as its value.
- 5) Once, every node gets assigned a color, we print the dictionary.
- 6) So we can get the node number and in which colour it is painted at the end.

## Code:

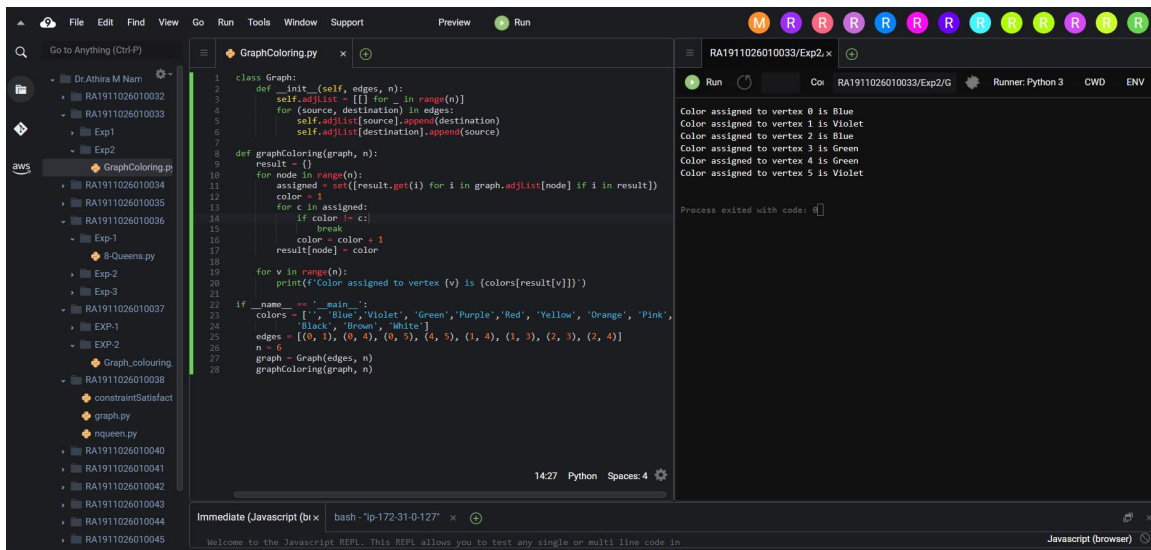
```
class Graph:
    def __init__(self, edges, n):
        self.adjlist = [[] for _ in range(n)]
        for (source, destination) in edges:
            self.adjlist[source].append(destination)
            self.adjlist[destination].append(source)

def graphColoring(graph, n):
    result = {}
    for node in range(n):
        assigned = set([result.get(i) for i in graph.adjlist[node] if i in result])
        color = 1
        for c in assigned:
            if color != c:
                break
            color = color + 1
        result[node] = color

    for v in range(n):
        print(f'Color assigned to vertex {v} is {colors[result[v]]}')

if __name__ == '__main__':
    colors = ['', 'Blue', 'Violet', 'Green', 'Purple', 'Red', 'Yellow', 'Orange', 'Pink',
              'Black', 'Brown', 'White']
    edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
    n = 6
    graph = Graph(edges, n)
    graphColoring(graph, n)
```

## Output :



```
1 class Graph:
2     def __init__(self, edges, n):
3         self.adjlist = [[] for _ in range(n)]
4         for (source, destination) in edges:
5             self.adjlist[source].append(destination)
6             self.adjlist[destination].append(source)
7
8     def graphColoring(self, graph, n):
9         result = {}
10        for node in range(n):
11            assigned = set()
12            for i in range(n):
13                if i in graph.adjlist[node]:
14                    assigned.add(i)
15            color = 1
16            for c in assigned:
17                if color == c:
18                    break
19            color = color + 1
20            result[node] = color
21
22        for v in range(n):
23            print(f'Color assigned to vertex (v) is {result[v]}')
24
25    if __name__ == '__main__':
26        colors = ['Blue', 'Violet', 'Green', 'Purple', 'Red', 'Yellow', 'Orange', 'Pink',
27                'Black', 'Brown', 'White']
28        n = 6
29        edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
30        graph = Graph(edges, n)
31        graphColoring(graph, n)
```

Color assigned to vertex 0 is Blue  
Color assigned to vertex 1 is Violet  
Color assigned to vertex 2 is Blue  
Color assigned to vertex 3 is Green  
Color assigned to vertex 4 is Green  
Color assigned to vertex 5 is Violet

Process exited with code: 0

## Observation :

1) By using greedy method we can traverse the whole graph. And then we can colour the nodes in just traversing the graph once.

2) Here time complexity is  $O(n * m)$

Here  $n$  is number of nodes

$M$  is number of edges

## RESULT:

Graph Coloring program is successfully implemented