

Prajwal Gupta

RA1911003010660

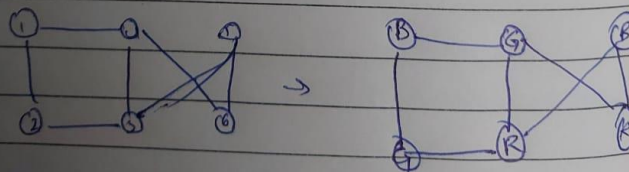
AI LAB-2 Graph Coloring

Aim- Implementation of Real word problems- Graph Coloring.

Problem formulation → Graph coloring is a way of coloring a graph's vertices such that no two adjacent vertices share the same color.

Problem solving → The goal is to color the vertices of the graph such that adjacent vertices have different colors. Greedy coloring considers the vertices of the graph in sequence and assigns each vertex its first available color i.e. vertices are considered in a specific order v_1, v_2, \dots, v_n and v_i is assigned the smallest available color which is not used by any of v_i 's neighbours.

eg-



Algorithm →

- Make a class to represent a graph object
- Function to assign colors to vertices of graph
- Make a set to keep track of the color assigned to each vertex.
- assign colors one by one
 - check colors of adjacent vertices of 'u' and store them in a different set
 - check for first free color
 - assign vertex 'u' the ~~same~~ first available color.

Code-

class Graph:

```
def __init__(self, edges, N):
```

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

```
    for (src, dest) in edges:
```

```
        self.adj[src].append(dest)
```

```
self.adj[dest].append(src)
```

```
def colorGraph(graph):
```

```
    result = {}
```

```
    for u in range(N):
```

```
        assigned = set([result.get(i) for i in graph.adj[u] if i in result])
```

```
        color = 1
```

```
        for c in assigned:
```

```
            if color != c:
```

```
                break
```

```
        color = color + 1
```

```
    result[u] = color
```

```
    for v in range(N):
```

```
        print("Color assigned to vertex", v, "is", colors[result[v]])
```

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

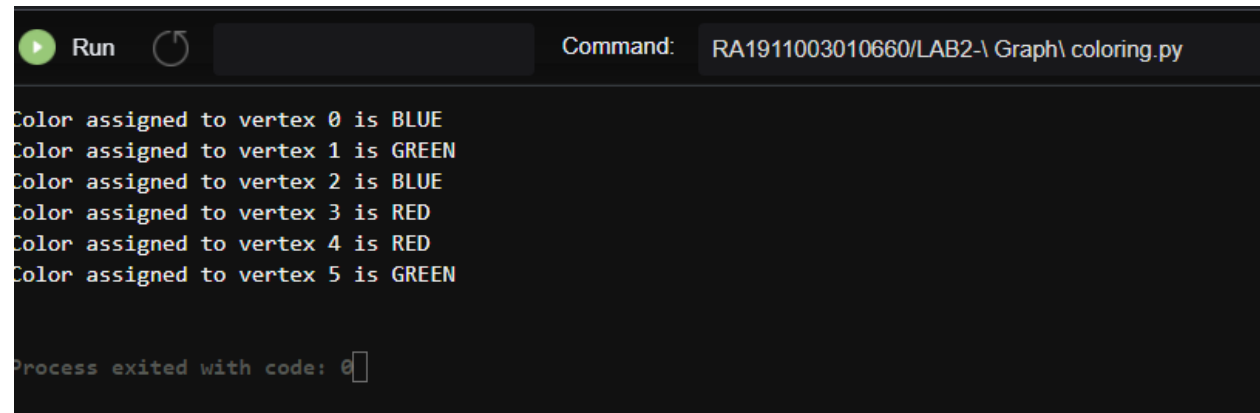
```
edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
```

N = 6

```
graph = Graph(edges, N)
```

```
colorGraph(graph)
```

Output-



The screenshot shows a code execution interface with a dark background. At the top, there is a 'Run' button with a green play icon and a circular refresh icon. To the right of these icons is a text field containing the command path 'RA1911003010660/LAB2-\ Graph\ coloring.py'. Below this header, the output of the program is displayed in a monospaced font. The output consists of six lines, each indicating the color assigned to a specific vertex. The colors are: vertex 0 is BLUE, vertex 1 is GREEN, vertex 2 is BLUE, vertex 3 is RED, vertex 4 is RED, and vertex 5 is GREEN. At the bottom of the output, it says 'Process exited with code: 0' followed by a small square icon.

```
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

Process exited with code: 0
```