

SUPREET GANGULY

RA1811003010984

EXP 3- CSP Problem

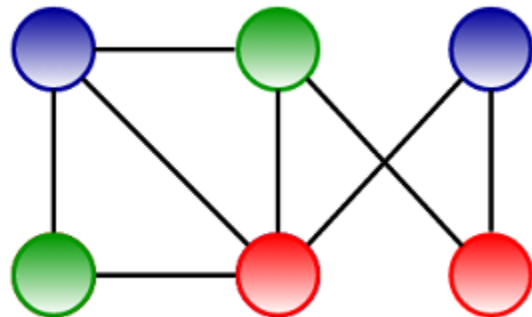
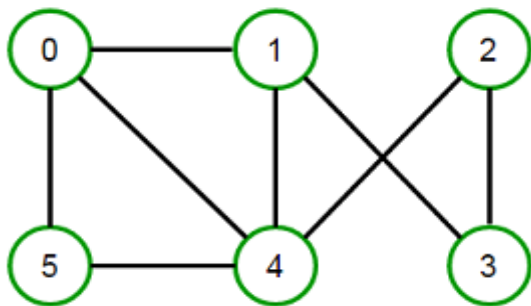
PROBLEM: (Graph Colouring Problem)

Graph coloring problem is a special case of graph labeling. In this problem, each node is colored into some colors. But coloring has some constraints. We cannot use the same color for any adjacent vertices. For solving this problem, we need to use the greedy algorithm.

Applications of Graph Coloring :

- 1) Making Schedule or Time Table
- 2) Mobile Radio Frequency Assignment
- 3) Map Coloring
- 4) Register Allocation

DIAGRAM FOR UNDERSTANDING



CODE

```
# 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", "VOILET"]
```

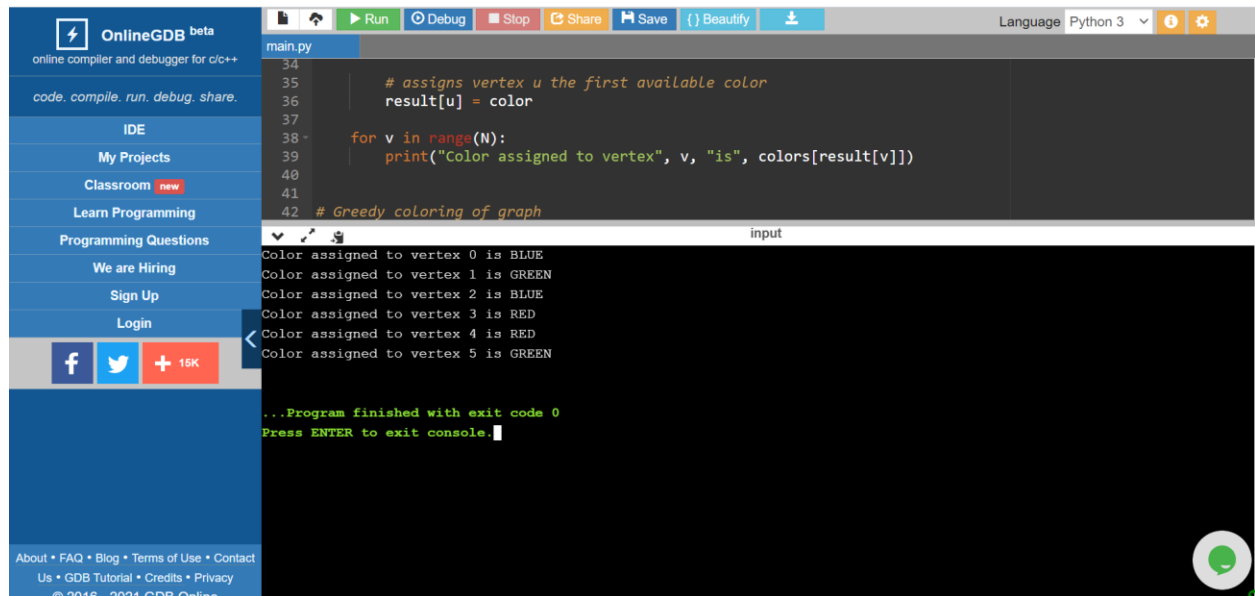
```
    # of graph edges as per above diagram
    edges = [(0, 1), (0, 4), (0, 5), (4, 5), (1, 4), (1, 3), (2, 3), (2, 4)]
```

```
    # 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



The screenshot displays the OnlineGDB beta web interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, We are Hiring, Sign Up, and Login. Below these are social media icons for Facebook, Twitter, and a '+ 16K' button. The main area shows a code editor with a file named 'main.py' containing Python code for a greedy graph coloring algorithm. The code includes comments and a loop that assigns colors to vertices. Below the editor is a console window showing the program's output, which lists the color assigned to each of the six vertices. The program ends with a message indicating it finished with exit code 0.

```
main.py
34
35     # assigns vertex u the first available color
36     result[u] = color
37
38     for v in range(N):
39         print("Color assigned to vertex", v, "is", colors[result[v]])
40
41
42     # Greedy coloring of graph

input
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

...Program finished with exit code 0
Press ENTER to exit console.
```

RESULT

The CSP problem of graph colouring is successfully executed using python language.