118. Graph Coloring

Code:

```
def is_valid(graph, v, color, c):
    for i in range(len(graph)):
        if graph[v][i] == 1 and color[i] == c:
            return False
    return True

def graph_coloring_util(graph, m, color, v):
    if v == len(graph):
        return True

    for c in range(1, m + 1):
        if is_valid(graph, v, color, c):
            color[v] = c
            if graph_coloring_util(graph, m, color, v + 1):
                return True
            color[v] = 0

    return False

def graph_coloring(graph, m):
    color = [0] * len(graph)
    if graph_coloring_util(graph, m, color, 0):
        return color
    else:
        return None

graph = [
    [0, 1, 1, 1],
    [1, 0, 1, 0],
```

```
    [1, 1, 0, 1],
    [1, 0, 1, 0]
]
m = 3
result = graph_coloring(graph, m)
if result:
    print("Solution exists: Following are the assigned colors:")
    print(result)
else:
    print("No solution exists")
```

output:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/problem.py
Solution exists: Following are the assigned colors:
[1, 2, 3, 2]
PS C:\Users\karth>
```

Time complexity:$f(n)=o(m^n)$