Assignment 2: Graph Colouring Problem

Team Members:

Ahmet Bozbay 150119861

Onur Kaya 150119860

Ubeydullah Fikri Günay 150117063

Algorithm:

1. For finding the optimal number of colours we start by colouring the first vertex with first colour.
2. Do following with remaining V-1 vertices.
    a. Colour the given vertex with the lowest colour which have not been assigned to its neighbouring vertices and if all colours appeared on adjacent vertices, then assign it a new colour.

Validation:

This algorithm may not always give optimal colours but guaranteed to give d+1 if d is the degree of curve because in our algorithm if d is the maximum neighbours that a vertex has then by our algorithm, we will choose colour from 1 to d+1 if we start colouring from 1. This can easily be proved by induction.

Code explanation:

Here is the code snippet that allows us to create new vertices.

```
def add_edge(adj, v, w):
    adj[v].append(w)
    adj[w].append(v)
    return adj
```

This snippet belongs to part 2.a

```
color = 0
while color < v:
    if (available[color] == False):
        break

    color += 1

ans[u] = color

for i in adj[u]:
    if (ans[i] != -1):
        available[ans[i]] = False
```

The code snippets here take the inputs from input.txt and print the generated answers to output.txt as output.

```
file1 = open('output.txt',"w")
file1.write(str(max(ans)+1))
file1.write('\n')
for u in range(v):
    file1.write(str(ans[u]))
    file1.write(' ')
```

```
file = open('input.txt',"r")
lines = file.readlines()
line = lines[0]
x = line.split(' ')
v = int(x[1])
e = int(x[2])
g = [[] for i in range(v)]
for line in lines[1:]:
    x = line.split(' ')
    g = add_edge(g, int(x[1])-1, int(x[2])-1)

optimal_coloring(g, v)
```

Work distribution:

There was no specific work distribution. After meeting and researching together on discord, we reached the final version of the code. After that, we made certain optimizations. The report was written in the same way with the work of 3 of us.