

Ex No.2: Graph Colouring Problem

Code:

```
colors = ['Red', 'Green', 'Blue']

states = ['WA', 'NT', 'SA', 'Q', 'NSW', 'V', 'T']

neighbors = {}
neighbors['WA'] = ['NT', 'SA']
neighbors['NT'] = ['WA', 'SA', 'Q']
neighbors['SA'] = ['WA', 'NT', 'Q', 'NSW', 'V']
neighbors['Q'] = ['NT', 'SA', 'NSW']
neighbors['NSW'] = ['SA', 'Q', 'V']
neighbors['V'] = ['SA', 'NSW']
neighbors['T'] = []

colors_of_states = {}

def promising(state, color):
    for neighbor in neighbors.get(state):
        color_of_neighbor = colors_of_states.get(neighbor)
        if color_of_neighbor == color:
            return False

    return True

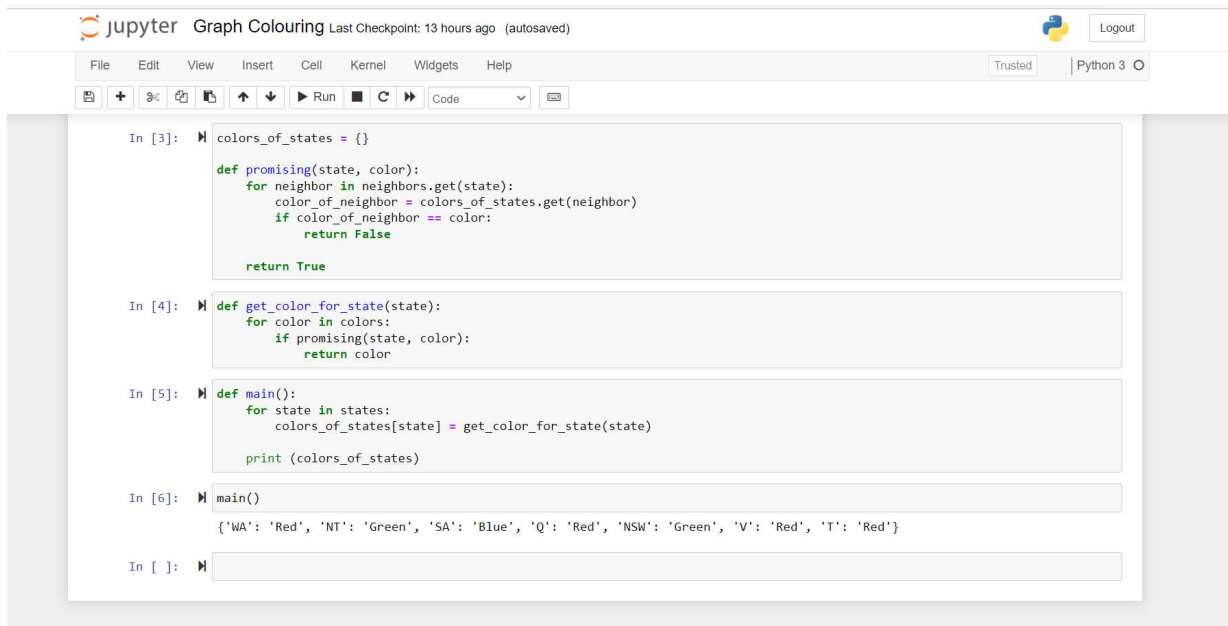
def get_color_for_state(state):
    for color in colors:
        if promising(state, color):
            return color

def main():
    for state in states:
        colors_of_states[state] = get_color_for_state(state)

    print (colors_of_states)

main()
```

Output:



The image shows a Jupyter Notebook interface with the title "Graph Colouring". The top bar indicates the last checkpoint was 13 hours ago and it is autosaved. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains five input cells with the following Python code:

```
In [3]: colors_of_states = {}

def promising(state, color):
    for neighbor in neighbors.get(state):
        color_of_neighbor = colors_of_states.get(neighbor)
        if color_of_neighbor == color:
            return False
    return True

In [4]: def get_color_for_state(state):
    for color in colors:
        if promising(state, color):
            return color

In [5]: def main():
    for state in states:
        colors_of_states[state] = get_color_for_state(state)

    print (colors_of_states)

In [6]: main()

{'WA': 'Red', 'NT': 'Green', 'SA': 'Blue', 'Q': 'Red', 'NSW': 'Green', 'V': 'Red', 'T': 'Red'}
```

The final cell is an empty input prompt.