**EXPERIMENT 11:**

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

neighbors = {

'WA': ['NT', 'SA'],

'NT': ['WA', 'SA', 'Q'],

'SA': ['WA', 'NT', 'Q', 'NSW', 'V'],

'Q': ['NT', 'SA', 'NSW'],

'NSW': ['SA', 'Q', 'V'],

'V': ['SA', 'NSW'],

'T': []

}

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

def is\_valid(region, color, assignment):

for neighbor in neighbors[region]:

if neighbor in assignment and assignment[neighbor] == color:

return False

return True

def backtrack(assignment):

if len(assignment) == len(regions):

return assignment

unassigned = [r for r in regions if r not in assignment]

region = unassigned[0]

for color in colors:

if is\_valid(region, color, assignment):

assignment[region] = color

result = backtrack(assignment)

if result:

return result

del assignment[region]

return None

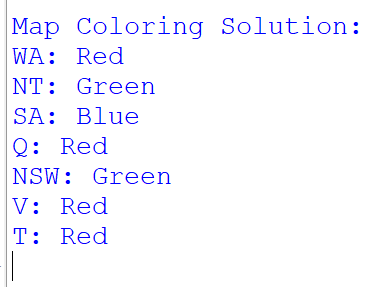
solution = backtrack({})

print("Map Coloring Solution:")

for region in regions:

print(f"{region}: {solution[region]}")

**OUTPUT:**

****