# Define the map as a dictionary of neighbors

australia\_map = {

'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': []

}

# Define possible colors

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

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

for neighbor in australia\_map[region]:

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

return False

return True

def backtrack(assignment):

# If assignment is complete

if len(assignment) == len(australia\_map):

return assignment

# Select an unassigned region

unassigned = [r for r in australia\_map if r not in assignment]

region = unassigned[0]

for color in colors:

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

assignment[region] = color

result = backtrack(assignment)

if result:

return result

del assignment[region] # Backtrack

return None

# Start solving

solution = backtrack({})

print("Solution:", solution)

