27. Map Coloring using Constraint Satisfaction (CSP)

def is\_valid(state, country, color, neighbors):

for neighbor in neighbors[country]:

if state.get(neighbor) == color:

return False

return True

def backtrack(state, colors, countries, neighbors):

if len(state) == len(countries):

return state

country = [c for c in countries if c not in state][0]

for color in colors:

if is\_valid(state, country, color, neighbors):

state[country] = color

result = backtrack(state, colors, countries, neighbors)

if result:

return result

del state[country]

return None

# Example: Australia map coloring

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

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

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': []

}

solution = backtrack({}, colors, countries, neighbors)

print("Map Coloring Solution:", solution)

OUTPUT:

