Exercise 11

def is\_safe(region, color, assignment, neighbors):

for neighbor in neighbors[region]:

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

return False

return True

def backtrack(assignment, regions, colors, neighbors):

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

return assignment

region = [r for r in regions if r not in assignment][0]

for color in colors:

if is\_safe(region, color, assignment, neighbors):

assignment[region] = color

result = backtrack(assignment, regions, colors, neighbors)

if result:

return result

del assignment[region]

return None

# Example map: regions and their neighbors

neighbors = {

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

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

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

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

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

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

'T': []

}

regions = list(neighbors.keys())

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

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

print(solution)

