Map Coloring Problem using CSP (Backtracking)

# Python Code

# Define the map and colors  
regions = ['WA', 'NT', 'SA', 'Q', 'NSW', 'V', 'T']  
adjacency = {  
 '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']  
  
# Backtracking CSP Solver  
def is\_valid(region, color, assignment):  
 for neighbor in adjacency[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  
  
# Solve the map coloring CSP  
solution = backtrack({})  
  
# Output  
print("Map Coloring Solution:")  
for region in regions:  
 print(f"{region}: {solution[region]}")

# Output

Map Coloring Solution:  
WA: Red  
NT: Green  
SA: Blue  
Q: Red  
NSW: Green  
V: Red  
T: Red