```python
def is_safe(region, color, assignment, neighbors):
    for neighbor in neighbors.get(region, []):
        if neighbor in assignment and assignment[neighbor] == color:
            return False
    return True

def map_coloring(regions, colors, neighbors, 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_safe(region, color, assignment, neighbors):
            assignment[region] = color
            result = map_coloring(regions, colors, neighbors, assignment)
            if result:
                return result
            del assignment[region]

    return None
```

```python

regions = ['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': ['Q', 'SA', 'V'],
    'V': ['SA', 'NSW'],
    'T': []
}

solution = map_coloring(regions, colors, neighbors)

if solution:
    print("Map coloring solution:")
    for region in regions:
        print(f"{region}: {solution[region]}")
else:
    print("No solution found.")

```

```
>>> %Run -c $EDITOR_CONTENT

  Map coloring solution:
  WA: Red
  NT: Green
  SA: Blue
  Q: Red
  NSW: Green
  V: Red
  T: Red

>>>
```