CSE-AI TY A div

Student Name: Prachee Prasad

Roll No.: 381060

Assignment 2

Implement Constraint Satisfaction Problem (Forward Checking Algorithm – Australia Map Coloring)

Problem Statement:

To implement a **Constraint Satisfaction Problem (CSP)** using the **Forward Checking Algorithm** for the **Australia Map Coloring Problem**, where each region must be colored such that no two adjacent regions have the same color.

Objective:

To understand how **constraint propagation** and **forward checking** help reduce the search space in CSPs by eliminating inconsistent domain values before exploring further assignments.

Requirements:

- Programming Language: **C++ / Python**
- Concepts: Constraint Satisfaction Problems, Backtracking, Forward Checking, Domain Filtering
- Input: Regions and their adjacency list
- Output: Valid coloring of each region such that no two neighbors share the same color

Operating System:

Windows / Linux / macOS

Libraries and Packages Used:

- **C++ STL** (`map`, `vector`, `set`)
- No external libraries required

Theory:

Definition:

A **Constraint Satisfaction Problem (CSP)** consists of:

- A set of **variables** (e.g., regions on a map)
- A **domain** of possible values for each variable (e.g., colors)
- A set of **constraints** that specify allowable combinations of values

The **Forward Checking Algorithm** is a constraint propagation technique that, after assigning a variable, looks ahead to eliminate inconsistent values from the domains of unassigned variables.

Structure:

- **Variables:** {WA, NT, SA, Q, NSW, V, T}
- **Domains:** {Red, Green, Blue}
- **Constraints:** Adjacent regions must not share the same color.

Methodology:

1. Select an unassigned variable (region).
2. Assign a color from its domain.
3. Apply **forward checking** — remove that color from adjacent regions' domains.
4. If any domain becomes empty → backtrack.
5. Continue until all regions are assigned valid colors or no solution exists.

Advantages:

- Reduces the search space by early detection of conflicts.
- Faster than simple backtracking due to domain pruning.
- Improves efficiency in CSPs with multiple constraints.

Limitations:

- May still require backtracking in complex problems.
- Does not guarantee global consistency (only local).
- Performance depends on variable ordering and domain size.

Working / Algorithm:

Algorithm Steps:

1. Input all variables (regions) and constraints (adjacencies).
2. Assign an initial color to a variable.
3. For each assignment:
    - Check consistency with previously assigned variables.
    - Apply forward checking — remove invalid colors from adjacent regions.
4. If a domain becomes empty → backtrack.
5. Repeat until all variables are assigned colors.
6. Output the valid coloring solution.

Conclusion:

The **Forward Checking Algorithm** efficiently solves the Australia Map Coloring problem by detecting conflicts early and reducing unnecessary searches. It demonstrates how **constraint propagation** enhances **backtracking search** in CSPs, providing an optimal and consistent coloring solution.