

Project 2 Report: CSP Graph Coloring

What we built

We built a CSP solver for graph coloring in `main.py`. The solver reads a graph file, assigns a color to each vertex, and makes sure no edge connects two vertices of the same color.

Input handling

The parser reads: - comment lines that start with `#` - a color line like `colors = 4` - edge lines like `u,v`

The parser ignores self-loops. It also removes duplicate undirected edges by storing each edge in sorted order.

CSP method used

The solver uses backtracking search with: - `MRV` to pick the next variable - a degree-based tie break inside `MRV` - `LCV` to order color values - `AC-3` for constraint propagation

At each step, the code: 1. picks one unassigned vertex with MRV, 2. tries colors in LCV order, 3. runs AC-3 after each trial assignment, 4. backtracks if a domain becomes empty.

If all vertices get valid colors, the solver returns the assignment. If no valid coloring exists, it returns `None`.

Test work done

`test.py` includes unit tests for: - file parsing (comments, duplicates, self-loops), - graph build, - `revise`, - `ac3`, - `select_mrv`, - `order_lcv`, - full `solve` cases.

The solve tests cover: - simple solvable graphs, - 2-color and 3-color cases, - complete graph coloring, - one unsolvable case.

Files

- `Project2/main.py` : CSP solver implementation
- `Project2/test.py` : unit tests