

Overview:

- Create a Constraint Satisfaction Problem (CSP) Solver for the following puzzles.
 - The basic 9x9 game, with demonstrations of solving an easy, medium, and hard puzzle found on the following website. <https://sudoku.com/>
- General details about the game can be found here: https://en.wikipedia.org/wiki/Glossary_of_Sudoku
- Provide a general encoding mechanism using a configuration file to provide the puzzle encoding.
- The same CSP code should be able to solve the problem generically for all attempted puzzles.
 - Thus, there should not be special program code for each of the differently encoded problems.

Experimental Procedure:

- Problems should come from the identified website (<https://sudoku.com/>) or puzzle books, this will ensure the problem is solvable.
- Create a plain Depth-First Search to determine if a solution can be found for your puzzles. If the solver takes more than an hour on your machine, you can provide the data you have obtained at termination.
- Apply your CSP implementation to solve three easy puzzles, three medium puzzles, and three hard puzzles; each with the following two techniques.
 - Variable/value ordering, using the minimum-remaining-values (MRV).
 - Variable/value ordering, using the least-constraining-value.
- It is optional: attempt some of the puzzles with AC-3 preprocessing before the Backtracking algorithm.

Coding Requirements:

- The program should be entirely driven by a configuration file, using no command-line input once the program is invoked.
- Log files should be used to provide output from the program.
- The submitted program must be run from the shell and cannot rely on a development environment to run it. A **Make** program can be used to start program execution and it would be helpful for the **Makefile** to provide each of the different ways the program can be run.
- A programming language of your choice can be used. The instructor needs to know how to install the execution environment on their machine if you are using an environment other than: python, java, or C++.
- The entire project will be contained in a directory with your name on it that has been compressed into a zip or tar file. It is preferred you do not use any other compression formats.
 - The compressed file will not contain any executable files.

Report Requirements:

The report will have the following sections:

- Program design explaining how the software is organized.
- User manual, explaining how the software is compiled, execution is started, and what to be expected for the input/output files.
- Design description how you are encoding your puzzles so that your generic CSP can function for all puzzle types.
- An explanation of the agent performance with results provided in charts/tables to characterized how your agent performs for different configurations.
- Summary of your results explaining why the agent produced the results it did.

The report will be included in the compressed directory and will be in PDF format.

Submission: Submission will be done via the course Moodle.