**Homework 6**

**CS 4750/7750 - Artificial Intelligence**

[Isaac Sexe](https://umsystem.instructure.com/groups/7438/users/79423) & Kaitlyn Zahn

***CSP FORMULATION & IMPLEMENTATION DESCRIPTION***

This program solves a 9x9 Sudoku puzzle using backtracking with an MRV and degree heuristic together with forward checking.

The program begins with the three instances given as 3D arrays. Each array contains the locations for all of the 1s on the game board, then the twos, and so on. The timer begins and the sudoku function calls populate\_board. This function loops through the given 3D arrays and assigns each value to the appropriate location. The original game board is printed and then forward\_checking is called.

The forward\_checking function iterates through the entire game board checking whether locations are available or full. If the position on the board is available, the isSafe function checks the current row, column, and 3x3 box to see what values could potentially go into that board. The forward\_checking function then adds each potential value to a domain list in the location of the board. After this, populate\_board is able to return whether the given board is solvable or not.

If the board is solvable, the backtracking function is called. If not, the program terminates and the user is notified that the puzzle cannot be solved. When the board can be solved, though, the backtracking function checks to see if the board is full already. If the board is finished, it changes each remaining domain list that has only one remaining option and inserts that value into the board. Otherwise, if the puzzle is not finished, the mrv function is called.

The mrv function iterates through the game grid and finds the position with the smallest remaining domain (the minimum remaining valid moves). It creates a list with all of the locations on the game board with the lowest domain size. Then, to break the tie, it calculates the degree heuristic. This is done by calculating the number of empty positions in the given move’s row, column, and 3x3 box. Finally, the move is selected.

Forward checking and backtracking are called recursively to re-evaluate the board each time a new move is made as domains and degrees change with each move.

Each instance’s original game board and solution are printed using the print\_board function which iterates through the board and adds formatting to make the solution readable to the user.

***REFERENCES***

Our code is referenced from https://www.geeksforgeeks.org/sudoku-backtracking-7/ and https://stackoverflow.com/questions/1557571/how-do-i-get-time-of-a-python-programs-execution as referenced in our code.

***INSTANCE A***

***Step 1***

*Variable Selected:* [4, 5]

*Domain Size:* 1

*Degree:* 9

*Value Assigned:* 7

***Step 2***

*Variable Selected:* [7, 5]

*Domain Size:* 1

*Degree:* 45

*Value Assigned:* 8

***Step 3***

*Variable Selected:* [8, 5]

*Domain Size:* 1

*Degree:* 44

*Value Assigned:* 1

***Step 4***

*Variable Selected:* [5, 5]

*Domain Size:* 1

*Degree:* 33

*Value Assigned:* 3

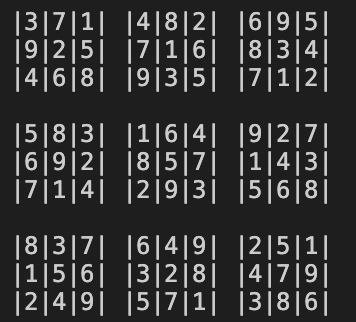
***Step 5***

*Variable Selected:* [4, 4]

*Domain Size:* 1

*Degree:* 23

*Value Assigned:* 5



***Solution***

***CPU Execution Time***

0.10408163070678711 seconds

***INSTANCE B***

***Step 1***

*Variable Selected:* [2, 5]

*Domain Size:* 2

*Degree:* 44

*Value Assigned:* 5

***Step 2***

*Variable Selected:* [2, 4]

*Domain Size:* 1

*Degree:* 12

*Value Assigned:* 8

***Step 3***

*Variable Selected:* [2, 7]

*Domain Size:* 2

*Degree:* 81

*Value Assigned:* 4

***Step 4***

*Variable Selected:* [2, 3]

*Domain Size:* 2

*Degree:* 82

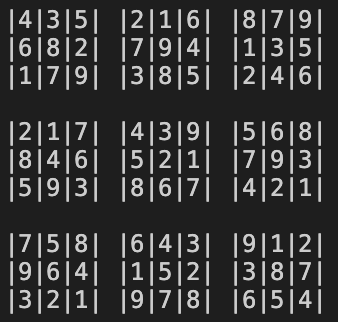
*Value Assigned:* 3

***Step 5***

*Variable Selected:* [2, 1]

*Domain Size:* 1

*Degree:* 10

*Value Assigned:* 7

***Solution***

***CPU Execution Time***

0.34627223014831543 seconds

***INSTANCE C***

***Step 1***

*Variable Selected:* [8, 6]

*Domain Size:* 2

*Degree:* 69

*Value Assigned:* 3

***Step 2***

*Variable Selected:* [8, 1]

*Domain Size:* 1

*Degree:* 12

*Value Assigned:* 4

***Step 3***

*Variable Selected:* [8, 8]

*Domain Size:* 2

*Degree:* 128

*Value Assigned:* 2

***Step 4***

*Variable Selected:* [8, 5]

*Domain Size:* 2

*Degree:* 154

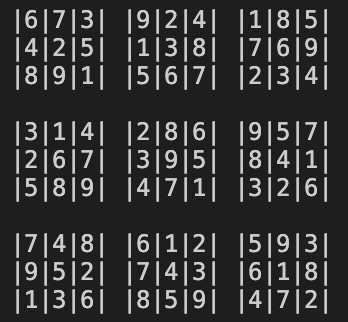
*Value Assigned:* 8

***Step 5***

*Variable Selected:* [8, 3]

*Domain Size:* 1

*Degree:* 10

*Value Assigned:* 9

***Solution***

***CPU Execution Time***

0.665522575378418 seconds