Sania Islam & Ella Reamer

Professor Bankston

CMPS 1100

23 Sept 2024

Midterm Checkpoint1

Repository:

https://github.com/CMPS1100-Sania-Ella-Final-Project/CMPS1100_FinalProject.git

Summary:

The current code prints a solved version of the Sudoku puzzle given when there is an existing solution. The driver code prints the solved grid if the solveSudoku function (starting at 0, 0) returns true, and prints "no solution exists" otherwise. The isSafe function checks that no number (besides 0) repeats in any given row, column, or 4x4 square. The solveSudoku function checks if the given position has a number greater than 0. If there is already a non-zero number in that position, it returns solveSudoku in the same grid and row, but one column over. If the program reaches the 16^{th} column in that given row, it returns to the 0^{th} column in the following row. If the program reaches the 15^{th} row and 16^{th} column (in a 0 indexed matrix), the function returns true, and prints the given matrix. However, if the solveSudoku function finds a position within the grid assigned 0, it uses the isSafe function to assign a safe number to that position, moves one column over, and then reassigns those positions 0 if isSafe returns false.

To do:

I need to change the code so that the grid, and preferably the size of the board, is user input. I would like to have a less tedious way to input the sudoku puzzle to be solved (maybe some kind of GUI?). I still need to test and debug it a lot. It likely is too inefficient for more complex puzzles.

Bugs & Obstacles:

When I input more difficult sudoku puzzles, the program just keeps running. I think there is either some infinite loop somewhere, or I need to optimize it because it is just too slow. Inputting a 16x16 sudoku puzzle as nested lists is also incredibly slow. It is very easy to mix up the positions and make your sudoku unsolvable but very time-consuming to check the input line-by-line.