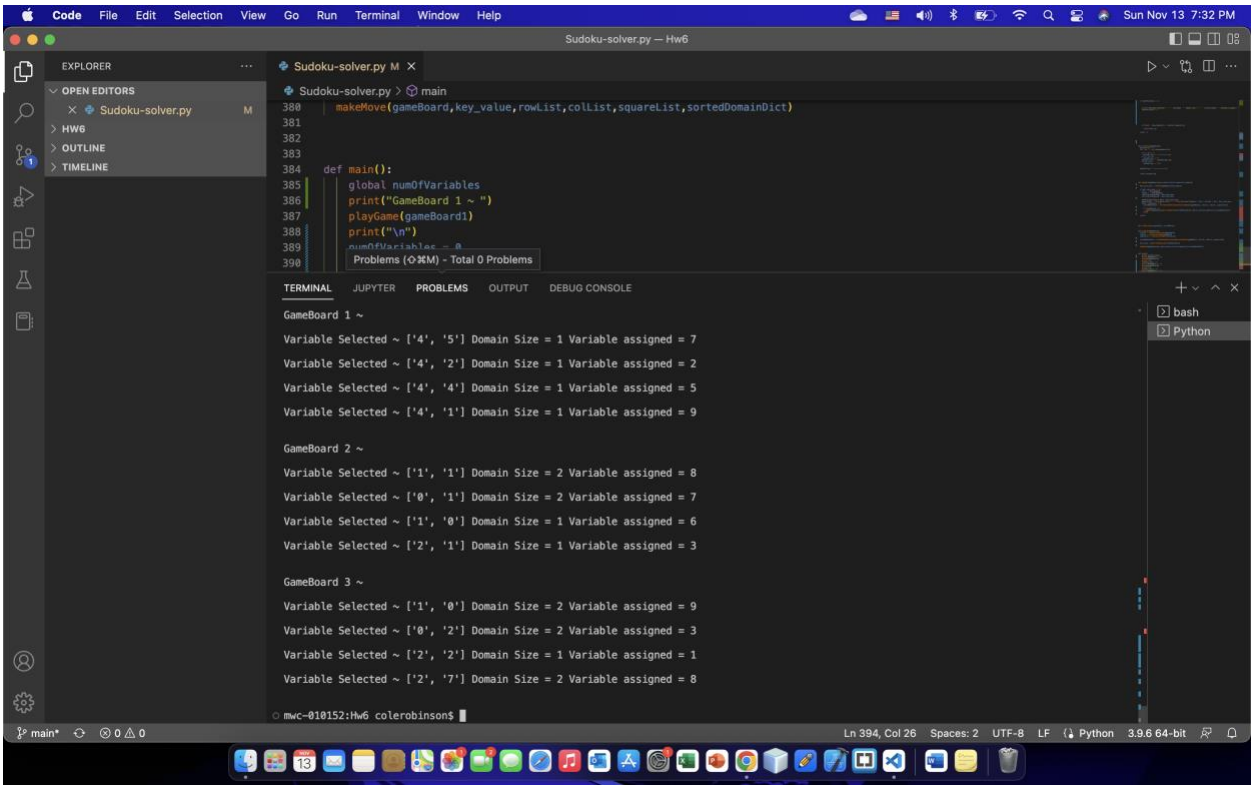
Hw6 – Sudoku Report

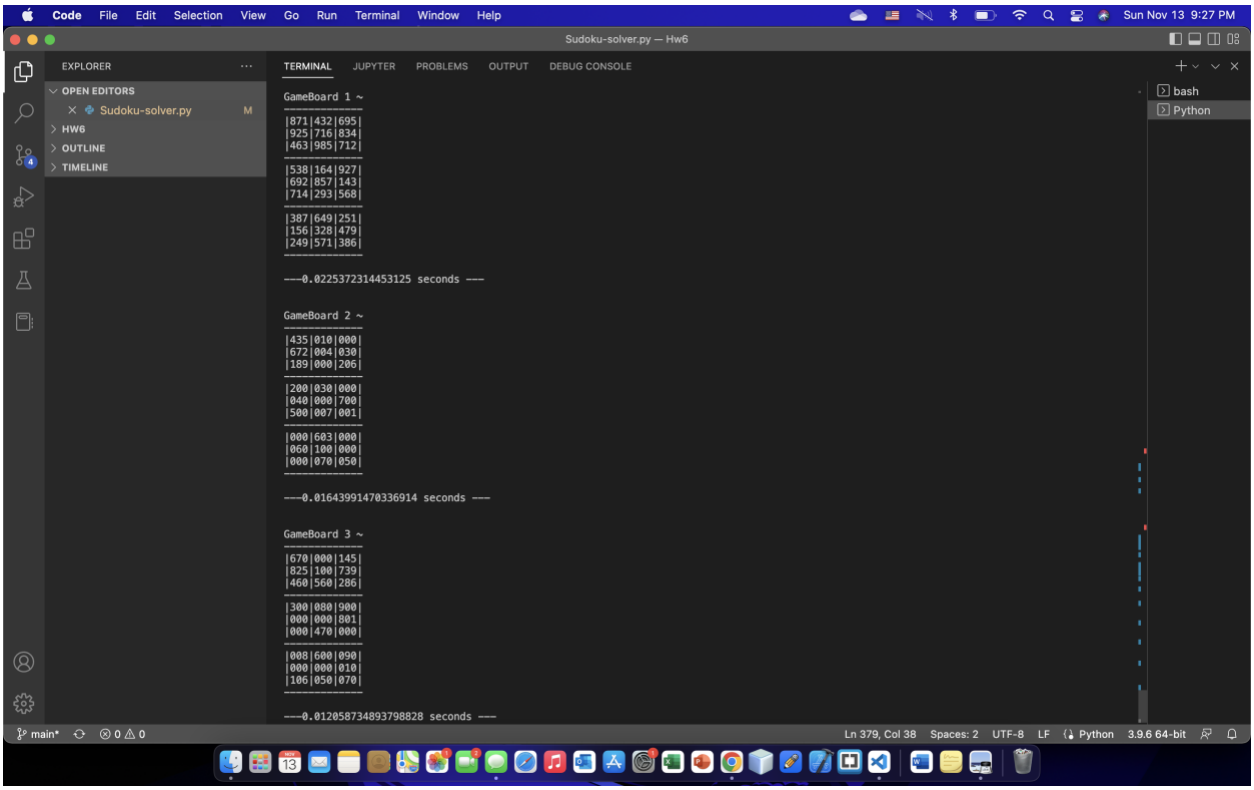
By: HW #6 Group 27 – Cole Robinson, Joel Hilliard, Srishti Singh

Description of implementation –

In this project the game boards were lists of integers. From those gameboards we calculated a list of lists for each row, column, and sub square of the game board. From there we created a domain for each spot that was not already occupied on the board. This was the CSP part of the project where values were only added to the domain if they were not in the same row, column, or square as the desired point. This domain dictionary used the coordinate pairs for the keys and the domains for the values. After that the domain dictionary was sorted based on length of each domain. Sorting the domain dictionary implements the MRV part of the project because this will make the program select a value with Minimum Remaining Values in its domain first. They kept the order of left to right and top to bottom to aid in tie breaking. The program then checks to see if a value in the shortest domain is valid by making sure it doesn't eliminate any domain other than its own and then executes the value that accomplishes this. Unfortunately, this program does not implement the degree heuristic and that is what makes it fail for game boards 2 and 3. After testing the points it walks through and assigns the point, updating the gameboard and regenerating the lists and domains for the new board until the game is complete. To reiterate, the program is incomplete for games 2 and 3 but reaches an end goal for game 1.

Below are the images of the programs prints. The first image contains the first 4 variables locations, domain lengths, and value assignments. (As mentioned before this program doesn't utilize the degree heuristic so that is omitted in the print statements.) The second image contains the resulting gameboards. Game board 1 is completely filled out but game board 2 and 3 only make it a little way through.


Code File Edit Selection View Go Run Terminal Window Help
Sun Nov 13 7:32 PM
Sudoku-solver.py — Hw6
EXPLORER
OPEN EDITORS
Sudoku-solver.py M
HW6
OUTLINE
TIMELINE
Sudoku-solver.py M
Sudoku-solver.py > main
makeMove(gameBoard,key_value,rowList,colList,squareList,sortedDomainDict)
def main():
global numOfVariables
print("GameBoard 1 ~ ")
playGame(gameBoard1)
print("\n")
Problems (⇧⌘M) - Total 0 Problems
TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE
bash
Python
GameBoard 1 ~
Variable Selected ~ ['4', '5'] Domain Size = 1 Variable assigned = 7
Variable Selected ~ ['4', '2'] Domain Size = 1 Variable assigned = 2
Variable Selected ~ ['4', '4'] Domain Size = 1 Variable assigned = 5
Variable Selected ~ ['4', '1'] Domain Size = 1 Variable assigned = 9
GameBoard 2 ~
Variable Selected ~ ['1', '1'] Domain Size = 2 Variable assigned = 8
Variable Selected ~ ['0', '1'] Domain Size = 2 Variable assigned = 7
Variable Selected ~ ['1', '0'] Domain Size = 1 Variable assigned = 6
Variable Selected ~ ['2', '1'] Domain Size = 1 Variable assigned = 3
GameBoard 3 ~
Variable Selected ~ ['1', '0'] Domain Size = 2 Variable assigned = 9
Variable Selected ~ ['0', '2'] Domain Size = 2 Variable assigned = 3
Variable Selected ~ ['2', '2'] Domain Size = 1 Variable assigned = 1
Variable Selected ~ ['2', '7'] Domain Size = 2 Variable assigned = 8
mwc-010152:Hw6 colerobinson$
main* ⊗ 0 ⚠ 0
Ln 394, Col 26 Spaces: 2 UTF-8 LF Python 3.9.6 64-bit


Code File Edit Selection View Go Run Terminal Window Help
Sun Nov 13 9:27 PM
Sudoku-solver.py — Hw6
EXPLORER
OPEN EDITORS
Sudoku-solver.py M
HW6
OUTLINE
TIMELINE
TERMINAL JUPYTER PROBLEMS OUTPUT DEBUG CONSOLE
bash
Python
GameBoard 1 ~
871	432	695
925	716	834
463	985	712
538	164	927
692	857	143
714	293	568
387	649	251
156	328	479
249	571	386
---0.0225372314453125 seconds ---
GameBoard 2 ~
435	010	000
672	004	030
189	000	206
200	030	000
040	000	700
500	007	001
000	603	000
060	100	000
000	070	050
---0.01643991470336914 seconds ---
GameBoard 3 ~
670	000	145
825	100	739
460	560	286
300	080	900
000	000	801
000	470	000
008	600	090
000	000	010
106	050	070
---0.012058734893798828 seconds ---
main* ⊗ 0 ⚠ 0
Ln 379, Col 38 Spaces: 2 UTF-8 LF Python 3.9.6 64-bit