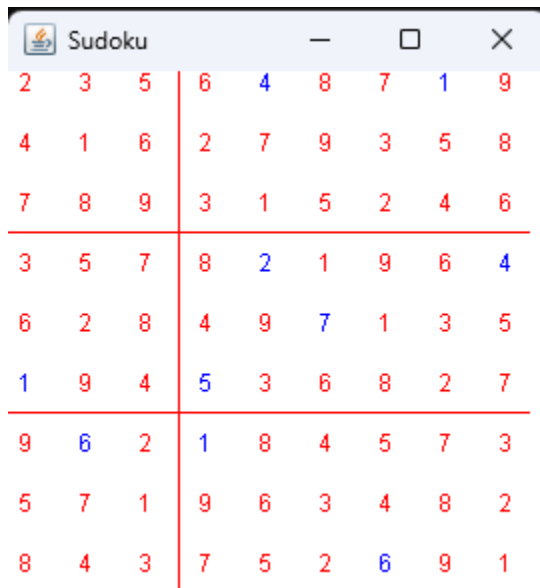**Title:** Solving Sudoku
Author: Eli Giglietti

Abstract:

In this project, we solved sudoku given a number of pre-filled squares. We used cells to hold the individual square's data, then a board to hold the cells and build the game, as well as find values given the ones we have. Then we used the sudoku class to solve the board by finding the valid values, and checking if they are correct. We used stacks, via a linked list, to store the values. Should a value not be correct, we would try a different one. Then if none worked, the value was popped off the stack, and the next item in the stack would then be changed to potentially find the rest of the values that would work. This pattern continued until there were either no valid solutions to the game or until there was a solution.

Results:

| Sudoku								
2	3	5	6	4	8	7	1	9
4	1	6	2	7	9	3	5	8
7	8	9	3	1	5	2	4	6
3	5	7	8	2	1	9	6	4
6	2	8	4	9	7	1	3	5
1	9	4	5	3	6	8	2	7
9	6	2	1	8	4	5	7	3
5	7	1	9	6	3	4	8	2
8	4	3	7	5	2	6	9	1

This is a solved sudoku board. The blue values are the beginning values that were locked. The stack wouldn't pop these values off. The red values are the ones that were added and taken out to find a correct solution.

Exploration:
1.

numLocked	numCorrect (out of 50)	numCorrect2	numCorrect3
10	49	50	50
20	40	41	30
30	1	1	0
40	0	0	0

There is a negative relationship between the number of original values and the number of correctly solved boards. As the number of original values goes up, the number of correct boards goes down.
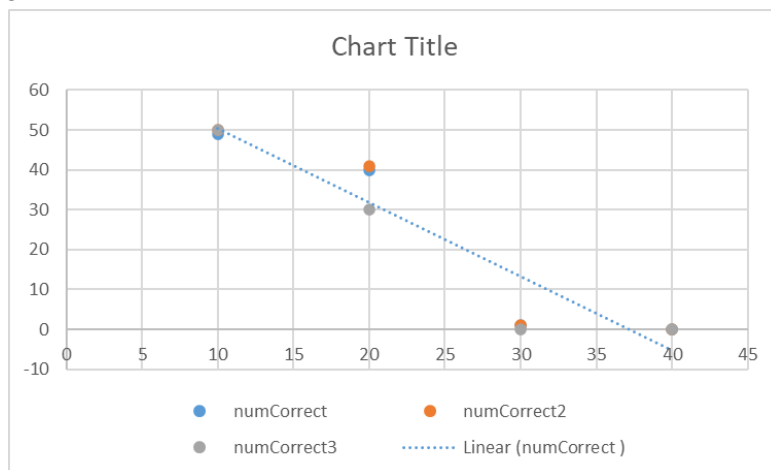
There is a negative trendline when plotting the values.

2.

Iterations of 10	Iterations of 20	Iterations of 30
108	409179	
15705	135	
38	170	
165	264746	
9129	2582	
123		
518		
34		
171		
56		
AVERAGES:		
2604.7	135362.4	

This is the number of calculations made to solve the board when given 10, 20, and 30 beginning values. When there were 30 given values, no board was being solved, along with 40 and so on. So any data for this would be misrepresentative of the question posed. However, there is a clear distinction between the number of calculations made when there are 10 given values and 20 given values, showing that a relationship between the two is likely.

Reflection:

Stacks are data structures where the first in is the last out. So, the "top" of the stack is the most recently added piece of data and is the only easily accessible one without sifting through the stack. This was helpful for sudoku, because it allowed you to only view and manipulate one cell at a time. If nothing works for the next value, the most recent value and that one only, would be changed. This allowed for checking one cell at a time in an ordered manner.

Worked With:
Liam Cotter
Ryan Mogauro
Henry Burdick