Adrian Gellert

Professor Maximillian Bender

CS231 Lecture A, Lab D

11 October 2022

Adrian's Project 4: Sudoku

ABSTRACT

In this project we created a stack-based Sudoku solving algorithm. The stack consisted of Cell's that held position values and an integer corresponding to a number valid within the rules of Sudoku. The project's goal was to gain practice implementing a depth-first search algorithm by creating this solving algorithm.

RESULTS

Figure 1: a solved board

Relationship between the number of randomly selected (but valid) initial values and the likelihood of finding a solution for the board (Based on number of boards that are solved per 10 runs of each number)	
Number of initial values	Percentage that have a result
10	100%
20	80%
30	10%
40	0%

Theoretically, with a greater number of fixed Cells, the stack containing unspecified Cells within the board would decrease. If this is the only thing that we think about, the runtime should decrease as the number of fixed cells increases. However, the number of times needed to

backtrack has to also be accounted for. I am not sure how to quantify the number of times that a given board will need to backtrack given its starting position.

REFLECTION

As stated in the abstract, this project used a stack made up of various Cells containing information about location and Sudoku values. Using a stack in this project allowed for the backtracking aspect of the Sudoku solving algorithm. That is, with a stack each node contains a separate package of information. Rather than having to copy the information within something like an array into an array of a different size, a stack allowed us to just move the top of the stack to the next node when backtracking, which in turn changes the size of the stack.

COLLABORATION

I collaborated with Catherine (“Jaime”) Yockey. I also received help with Professor Bender to fix errors in my code.