Cole Gotelli
CPSC 380
Hansen
10/23/2019

Project 1 - Sudoku Validator

The goal of this project was to make use of threads to check a filled Sudoku grid to find errors with its solution. The way my program is laid out is as follows. First the program takes one of the test files in as a command line parameter and converts the text file to a 2D array of integers. After the grid has been set up 3 threads are initialized: one for checking rows, one for checking columns, and one for calculating what the correct value should be. To start, the row checker compares every value to every other value in its row. If it finds 2 instances of the same value, it grabs the indices of the 2 values and puts them in a synchronous queue. The synchronous queue holds only a single value for practically no amount of time because as soon as something is added to it another section of code is notified that something has been added and takes it out of the queue. The column checker contains the code that takes the indices from the queue. Once the column thread has the 2 locations it checks the columns corresponding to those two locations to pinpoint which value actually needs to be changed. After this is determined the column checker sends that location to the fixer thread. The fixer thread first determines which subgrid the value exists in. Then it sums the values in the sub grid and subtracts it from the expected value and adds that value to the current incorrect value (I understand that this doesn't work when there is more than one error in a sub grid). Once all this data has been collected, all the threads join the main thread and the main thread outputs all the data.

Overall, this program works almost as intended. The program works perfectly for the first 4 test cases, however it runs into some issues with the last 2 test cases. For some reason beyond my comprehension, my program would only ever compute all the relevant data for one value even though it determined that there were multiple errors. In case 5, both the errors occurred in the same sub grid so that also caused issues for calculating the correct value for either value. I also recognize the fact that this particular solution relies on the fact that errors occur in both the rows and columns. Unfortunately, by the time I came up with a more robust (although maybe slightly less efficient) solution it was too late to change basically my whole codebase. If I did this assignment again I would check every row and every column, marking every index that might be wrong and then compare those indices to find to confirm false answers. As for determining which values would be correct I would use a boolean array to mark all the values that were found in the sub grid to highlight which values were missing.