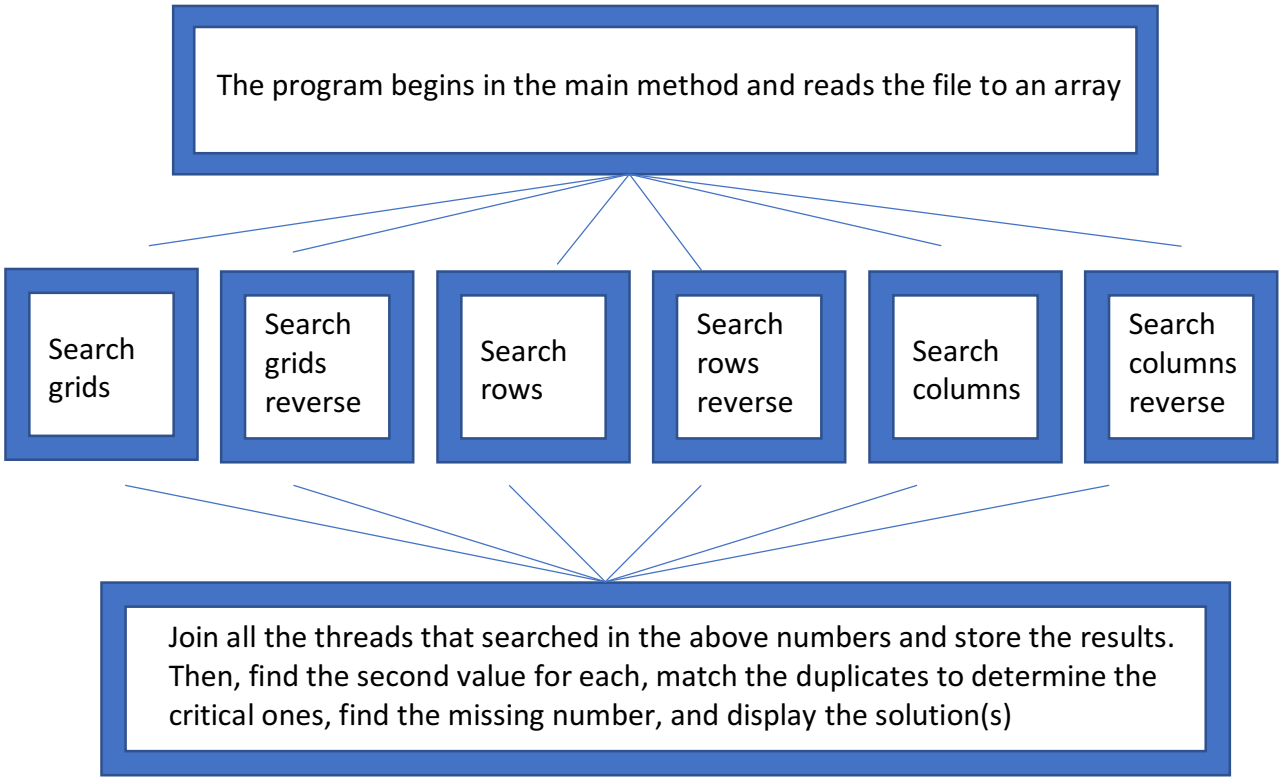
Ben Wasserman
March 7th, 2018
CPSC 380-01
Professor Hansen

Soduko Solution Validator

The design that I chose to implement can be shown with the diagram below. I will explain in a little bit more detail after the diagram.



In this program, there were 6 additional threads used, outside the main thread. I chose six threads because I wanted to ensure that each thread would take one part of what I thought would be the most computationally intensive part of the validator. The computationally intensive units were in determining what the duplicate values were. I split the threads as follows: two searching grids, two searching rows, and two searching columns. The second of each of these would search the grid, row, or column in the reverse direction from the previous one to try and minimize iterations. Then once these duplicate values were found, the threads were joined, additional duplicate values were found based on the initial six, and then I determined which duplicates needed to be changed based on the number of occurrences of each row and column. Lastly, the program searches for the missing number(s).