Solves Sudoku puzzle (No time to implement GUI grid-based input system)

* Code relies on manual input of 9x9 Sudoku matrix in test.m file
* Speed of code depends on puzzle difficulty

Algorithm used:

* Backtracking
  + Concept of filling in cells with only one solution first
  + Then goes through puzzle trying every other cell solution
* Recursively calls “solver”
  + Exits if all 81 cells filled
  + Otherwise, function runs until solution found
* “NewCheck” ensures current puzzle solvable

The overall time depends on the difficulty (tic toc implemented in test.m)

* ~ 8 seconds for “hard” puzzle, ~2 seconds for “easy” puzzle
* Tests conducted using online problem sets (from website link below)
  + (first puzzles in easy/hard difficulty sections tested)

For hard puzzle:

* “solver” function = 0.999 seconds
* “possibleNew” function took 3.438 seconds
* “gridSq” function = 3.545 seconds
* “NewCheck” function = 0.012 seconds

For easy puzzle, the specified functions took, in order:

* 0.238 seconds
* 0.874 seconds
* 0.912 seconds
* 0.009 seconds

Based on computer specifications, speed of solving decreased as computer specs decreased

* Tested using computer’s output power settings
  + Tested using “high power” and “low power” modes
  + Low power always took a few seconds more than high power

Hard puzzle ran:

* “solver” 688 times
* “possibleNew” 3065 times
* “gridSq” 248274 times

Easy puzzle ran:

* “solver” 153 times
* “possibleNew” 795 times
* “gridSq” 64404 times

(“Test”/“NewCheck” only runs once)

Memory usage:

* “solver” 8 KB
* “possibleNew” 4 KB
* “gridSq” 3 KB
* “NewCheck” 4 KB
* “Test” 1 KB

“Solver” goes through every possible solution

* Completely accurate
* Code not that long, but has recursive loops and complex for/while loops
  + Allows the code to be shorter while iterating through every possible solution.

No assumptions made

Constraints:

* Puzzle must be 9x9 Sudoku grid

<http://www.puzzles.ca/sudoku.html> used to test different difficulties of puzzles.

<http://www.techfinesse.com/game/sudoku_solver.php> used for algorithm (explained in “Description of Algorithm”).

1. Fills in single solution cells
2. Tries first solution in first cell
3. Recursively calls function
4. Repeats; if no possible solutions, will backtrack to previous solution and try new solution for that cell