Sudoku Puzzle Solver

For this sudoku puzzle coding project, I use the backtracking to solve the puzzle. The sudokuSolver () method will use the nested loops to test every cell’s possible numbers. When the nested loop detected an 0 from the matrix from the table, it will start the process to test the number for 1 to 9. The next step the if loop

If(checkrow(row, number)&&(checkCol(col, number)

&&(checkSquare(row,col,number)

will call checkrow, check col, and checkSquare to make check there is no repeated number from that row, column, and 3 by 3 square. After it’s safe the nested loop will place a number at the cell. The next if loop will call the sudoku solver again to test another cell. If the next sudoku solver fails the process, then it will be backtracking to the previous sudoku solver method to test another possible solution. The same process will continue until the whole matrix is solved.

The time complexity for this program should be O(9^(n\*n)) since each cell has 9 different solutions. Also, the time is dependent on how many 0 is on the table, so n\*n is the size of the problem and 9 is the numbers that will be used to test each cell.