As values entered must be integers, I was unable to remove 0 as an empty cell for input, as a blank space is not an integer. However, 0 was invalid as an option to solve the board.

Metrics:

Row Check-Total 10 times

Column Check-Total 10 times

SubGrid Check-Total 10 times

Total Calls between Row, Column, and Subgrid-30

Recursive Calls-Total 4 Times

Total Calls 4\*30= 120. While this may seem like a lot of work, the methods that are called more often are short and done quickly. Unfortunately, recursion is relatively inefficient, and is done in at worst O(n\*m)-n:rows, m:columns, as it goes through entire matrix often for error checking. Before adding the output file to display recursive solve attempt, the program ran quickly, often within 5 seconds. However, upon adding the output file, the program had to write and document each step, bogging down speed. With that said, having the method that takes the most time-calling numerous other methods, called less and shorter methods called more often made the program as efficient as possible due to the constraints.