Sudoku Solver (Backtracking 2) Java

Write a program to solve a Sudoku puzzle by filling the empty cells.

A sudoku solution must satisfy all of the following rules:

- 1. Each of the digits 1-9 must occur exactly once in each row.
- 2. Each of the digits 1-9 must occur exactly once in each column.
- 3. Each of the digits 1-9 must occur exactly once in each of the 9 3x3 sub-boxes of the grid.

The '.' character indicates empty cells.

Sample Input

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

board =

Sample Output

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	m	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

[["5","3","4","6","7","8","9","1","2"],["6","7","2","1","9","5","3","4","8"],["
1","9","8","3","4","2","5","6","7"],["8","5","9","7","6","1","4","2","3"],["4",
"2","6","8","5","3","7","9","1"],["7","1","3","9","2","4","8","5","6"],["9","6",
"1","5","3","7","2","8","4"],["2","8","7","4","1","9","6","3","5"],["3","4","5
","2","8","6","1","7","9"]]

Code

- a. StartingRow = 3*(row/3) & StartingCol = 3*(col/3)
- b. StartingRow = row row%3 & StartingCol = col col%3

Code

```
public boolean helper(char[][] board, int row, int col) {
        if(helper(board, nrow, ncol)) {
                if(helper(board, nrow, ncol))
```