Sudoku Solver

https://leetcode.com/problems/sudoku-solver/discuss/15752/Straight-Forward-Java-Solution-Using-Backtracking

```
class Solution {
    public void solveSudoku(char[][] board) {
        if (board == null || board.length == 0) return;
        solve(board);
    }

    public boolean solve(char[][] board) {
        // row, col, block
        for (int i = 0; i < 9; ++i)
            for (int j = 0; j < 9; ++j) {
                if (board[i][j] == '.') {
                    for (char c = '1'; c <= '9'; ++c) {
                        if (isValid(board, i, j, c)) {
                            board[i][j] = c;
                            if (solve(board)) return true;
                            board[i][j] = '.';
                        }
                    }
                    return false;
                }
            }
        return true;
    }

    private boolean isValid(char[][] board, int row, int col, char c){
        for(int i = 0; i < 9; i++) {
            if(board[i][col] != '.' && board[i][col] == c) return false; //check row
            if(board[row][i] != '.' && board[row][i] == c) return false; //check column
            if(board[3 * (row / 3) + i / 3][ 3 * (col / 3) + i % 3] != '.' &&
board[3 * (row / 3) + i / 3][3 * (col / 3) + i % 3] == c) return false; //check 3*3 block
        }
        return true;
    }
}
```