15. LC 37 Sudoku Solver

37. Sudoku Solver

Hard 5735 164 Add to List Share

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.

Example 1:

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

```
Input: board = [["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"]]
Output: [["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"]]
```

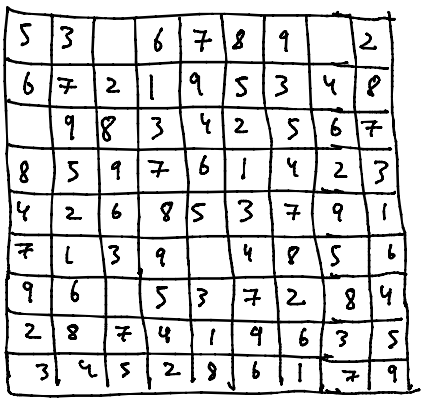
Explanation: The input board is shown above and the only valid solution is shown below:

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

9 x 9

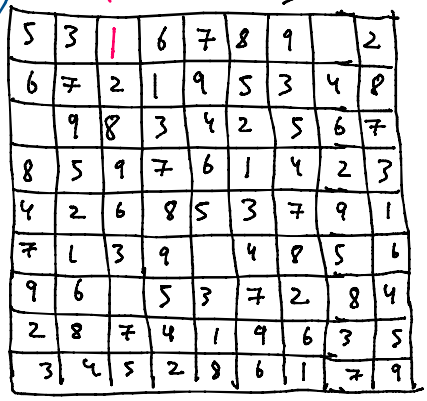
1) The digit 1–9 → once in every row
2) The digit 1–9 → once in every col
3) The digit 1–9 → once in every 3×3

} Rules

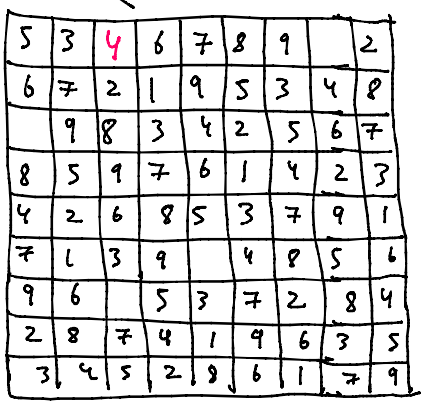


F

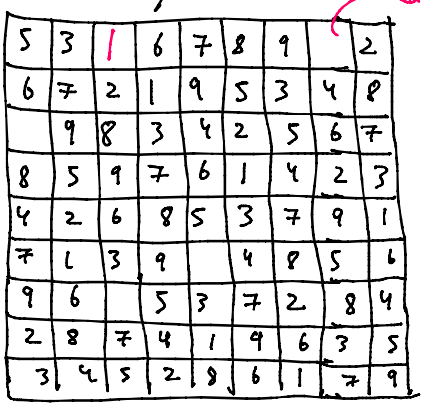
T



false



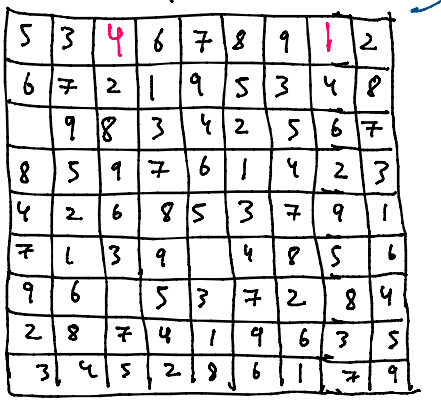
T



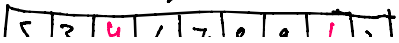
false

cannot fill anything

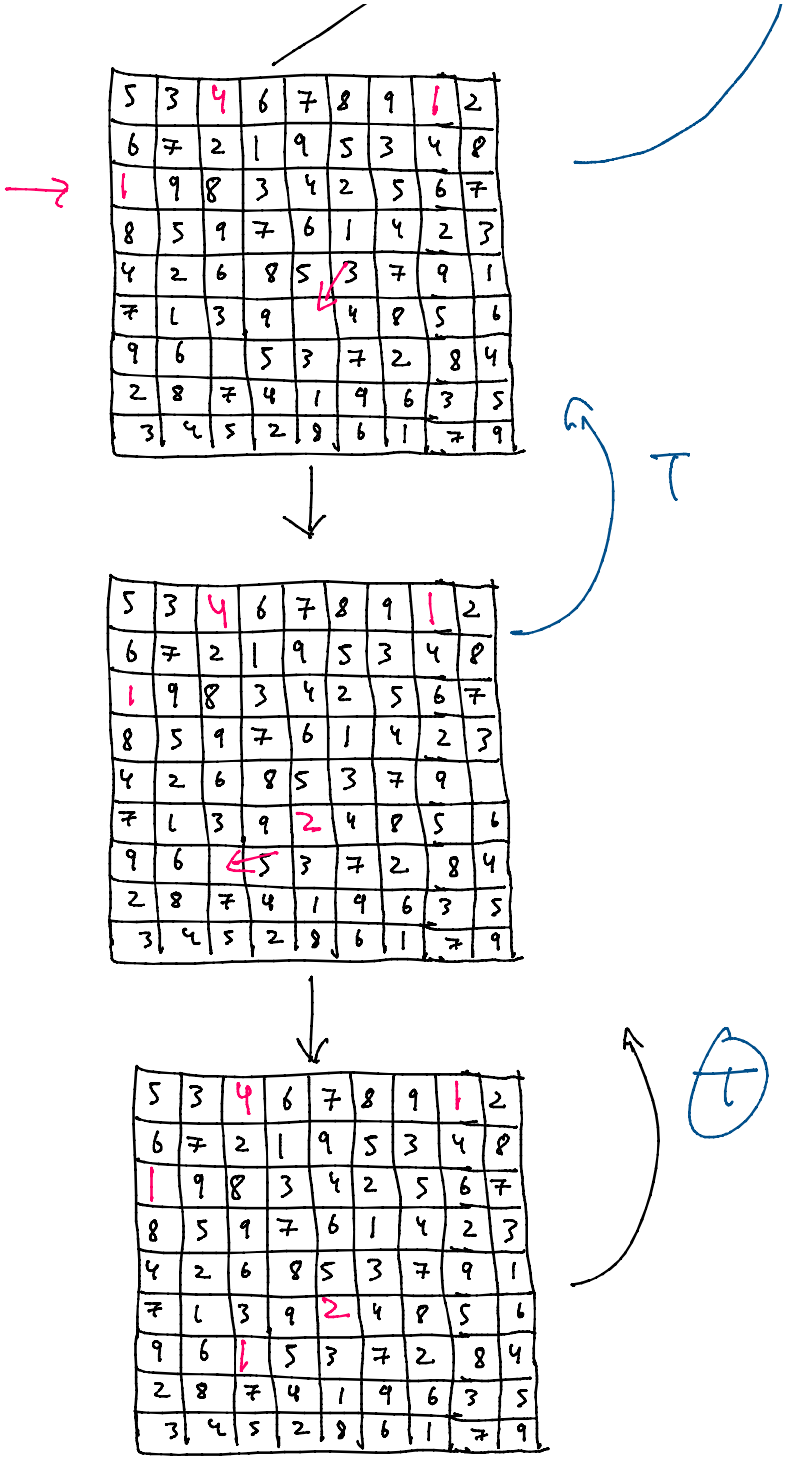
So return false



T



T



Approach :

1) Traverse the matrix and find the empty place

2) Once we find the empty place then we tried all the nos from 1 to 9 and check that it's a valid nos. or not by checking the rules.

3) And, we find the correct nos for that place than we find for the Second empty place in 9x9 matrix.

4) For Second empty place we repeat the Same process and if we doesn't get any nos, so we return false.

5) After getting false from solve (board) function we make all the places empty that we have filled than try for other member for first empty ace.

6) and after all we re ive calls, we got - ue, an have Stop ov v here nly an no need o care for r soln on

```
i C++    • Autocomplete

class Solution {
public:
    void solveSudoku(vector<vector<char>>& board) {
        solve(board);
    }

    bool solve(vector<vector<char>>& board){
        // finding the first empty so we traverse
        for(int i = 0; i < board.size(); i++){
            for(int j = 0; j < board[0].size(); j++){

                // if it empty
                if(board[i][j] == '.'){
                    //try out every possibility from 1 to 9
                    for(char c = '1'; c <= '9'; c++){
                        if(isValid(board, i, j, c)){
                            board[i][j] = c;

                            if(solve(board) == true)
                                return true;
                            else
                                board[i][j] ='.';   //remove it if false
                        }
```

```
            //for sub matrix
            if(board[3 * (row / 3) + i /3][3 * (col / 3) + i % 3] == c)
                return false;
        }
        return true;
    }
};
```

T.C ⇒ $O(9^{(n^2)})$ Worst case

S.C ⇒ $O(1)$