LeetCode Explore Day 24 Problems Mock new Contest Discuss Store September LeetCoding Challenge Premium

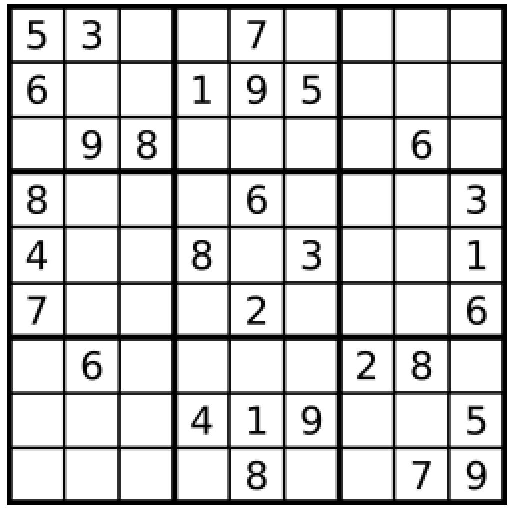
Description Solution Discuss (999+) Submissions

36. Valid Sudoku

Medium 1901 479 Add to List Share

Determine if a 9x9 Sudoku board is valid. Only the filled cells need to be validated **according to the following rules**:

1. Each row must contain the digits `1-9` without repetition.
2. Each column must contain the digits `1-9` without repetition.
3. Each of the 9 `3x3` sub-boxes of the grid must contain the digits `1-9` without repetition.



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

A partially filled sudoku which is valid.

The Sudoku board could be partially filled, where empty cells are filled with the character `'.'`.

Example 1:

```
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"]
]
Output: true
```

Problems Pick One Prev /59 Next

C++ Autocomplete

```
class Solution {
public:
    int giveblock(int i, int j){
        if(i<3){
            if(j<3)
                return 1;
            else if(j<6)
                return 2;
            return 3;
        }
        else if(i>=3 && i<6){
            if(j<3)
                return 4;
            else if(j<6)
                return 5;
            return 6;
        }
        else{
            if(j<3)
                return 7;
            else if(j<6)
                return 8;
            return 9;
        }

    }
    bool isValidSudoku(vector<vector<char>>& a) {
        unordered_map<int,vector<char>> rows;
        unordered_map<int,vector<char>> cols;
        unordered_map<int,vector<char>> blocks;

        for(int i=0;i<9;i++){
            for(int j=0;j<9;j++){
                if(a[i][j]!='.'){
                    if(find(rows[i].begin(),rows[i].end(),a[i][j])!=rows[i].end() ||
find(cols[j].begin(),cols[j].end(),a[i][j])!=cols[j].end() ||
find(blocks[giveblock(i,j)].begin(),blocks[giveblock(i,j)].end(),a[i][j])!=blocks[giveblock(i,j)].end())
                        return false;
                    rows[i].push_back(a[i][j]);
                    cols[j].push_back(a[i][j]);
                    blocks[giveblock(i,j)].push_back(a[i][j]);
                }
            }
        }
        return true;
    }
```

Your previous code was restored from your local storage. Reset to default

Console Contribute Run Code Submit