Homework3: Sudoku

You need to be familiar with the following topics to get the homework done:

- Basic python syntax
- Recursive function, such as `DFS` or `BFS`
- Object oriented programming
- Function decorator

Introduction

In this homework, you will need to design a sudoku solver that can automatically solve the sudoku puzzle within reasonable time constraint.

A sudoku puzzle is represented by a 9x9 matrix with some missing elements in it. Each element in the matrix is an integer value from 1 to 9.

```
For example:

        columns
                                  +-----+   +-----+   +-----+
   9,5,7,6,1,3,2,8,4              |9,5,7|   |6,1,3|   |2,8,4|
   4,8,3,2,5,7,1,9,6              |4,8,3|   |2,5,7|   |1,9,6|
r  6,1,2,8,4,9,5,3,7              |6,1,2|   |8,4,9|   |5,3,7|
o  1,7,8,3,6,4,9,5,2   ------->   +-----+   +-----+   +-----+
w  5,2,4,9,7,1,3,6,8   ------->   block0    block1    block2
s  3,6,9,5,2,8,7,4,1
   8,4,5,7,9,2,6,1,3              +-----+   +-----+   +-----+
   2,9,1,4,3,6,8,7,5              |1,7,8|   |3,6,4|   |9,5,2|
   7,3,6,1,8,5,4,2,9              |5,2,4|   |9,7,1|   |3,6,8|
                                  |3,6,9|   |5,2,8|   |7,4,1|
                                  +-----+   +-----+   +-----+
                                  block3    block4    block5

                                  +-----+   +-----+   +-----+
                                  |8,4,5|   |7,9,2|   |6,1,3|
                                  |2,9,1|   |4,3,6|   |8,7,5|
                                  |7,3,6|   |1,8,5|   |4,2,9|
                                  +-----+   +-----+   +-----+
                                  block6    block7    block8
```

A solved sudoku puzzle should satisfy following requirements:

1. Each row consists of a sequence of numbers from 1 to 9 ,and each digit can only occurs once
2. Each col consists of a sequence of numbers from 1 to 9 ,and each digit can only occurs once
3. Each block consists of a sequence of numbers from 1 to 9 ,and each digit can only occurs once

Requirements

Given the template code provided by TA, you need to fulfill all the methods and functions in the code. The score/point for each method or function is explained in the docstring.

MUST READ

- You cannot use any third-party package such as numpy, pandas, and etc.
- You can only use python primitive types and statements to solve the problem.
- Do not copy others code. (0 scores for punishment)

Expected Execution Result

`__init__`

```
# Content in sudoku2.txt
# ======================
# 0,5,7,6,1,3,2,8,4
# 4,8,3,2,5,0,0,0,6
# 6,1,2,8,4,0,5,0,7
# 1,7,8,3,0,0,0,0,2
# 5,2,4,9,7,1,3,6,8
# 3,6,0,0,0,0,7,4,1
# 8,4,5,7,9,2,6,1,3
# 2,9,1,4,3,6,8,7,5
# 7,3,6,1,8,0,0,0,0

# Test constructor
sudoku1 = Sudoku("sudoku2.txt")
```

executed in 131ms, finished 13:35:07 2020-08-24

`__str__`

```
# Test __str__ magic method
print(sudoku1, end="\n\n")
```

executed in 130ms, finished 13:35:07 2020-08-24

```
0 5 7 6 1 3 2 8 4
4 8 3 2 5 0 0 0 6
6 1 2 8 4 0 5 0 7
1 7 8 3 0 0 0 0 2
5 2 4 9 7 1 3 6 8
3 6 0 0 0 0 7 4 1
8 4 5 7 9 2 6 1 3
2 9 1 4 3 6 8 7 5
7 3 6 1 8 0 0 0 0
```

check_block

```
# Test check_block
print(sudoku1.check_block(0))
print(sudoku1.check_block(3))
print(sudoku1.check_block(6))
```

executed in 88ms, finished 13:35:07 2020-08-24

```
True
True
True
```

check_row

```
# Test check_row
print(sudoku1.check_row(0))
print(sudoku1.check_row(1))
print(sudoku1.check_row(2))
```

executed in 111ms, finished 13:35:07 2020-08-24

```
True
True
True
```

check_col

```
# Test check_col
print(sudoku1.check_col(0))
print(sudoku1.check_col(1))
print(sudoku1.check_col(2))
```

executed in 338ms, finished 13:35:07 2020-08-24

```
True
True
True
```

is_solved

```
# Test is_solved
print(sudoku1.is_solved())
```

executed in 94ms, finished 13:35:07 2020-08-24

```
False
```

solve

```
# Test solve
sudoku1.solve()

print(sudoku1, end="\n\n")
```

executed in 96ms, finished 13:35:07 2020-08-24

```
<function Sudoku.solve at 0x7f55d37e4158> executime time: 0.0010802745819091797
9 5 7 6 1 3 2 8 4
4 8 3 2 5 7 1 9 6
6 1 2 8 4 9 5 3 7
1 7 8 3 6 4 9 5 2
5 2 4 9 7 1 3 6 8
3 6 9 5 2 8 7 4 1
8 4 5 7 9 2 6 1 3
2 9 1 4 3 6 8 7 5
7 3 6 1 8 5 4 2 9
```

is_solved after solve

```
# Test is_solved
print(sudoku1.is_solved())
```

executed in 101ms, finished 13:35:08 2020-08-24

```
True
```