Graph-colorings: Sudoku game

esay hard seçsin önce gridi göster kullanıcıya, 2

APPLICATION

```
import numpy as np

grid = [[5,3,0,0,7,0,0,0,0],
        [6,0,0,1,9,5,0,0,0],
        [0,9,8,0,0,0,0,6,0],
        [8,0,0,0,6,0,0,0,3],
        [4,0,0,8,0,3,0,0,1],
        [7,0,0,0,2,0,0,0,6],
        [0,6,0,0,0,0,2,8,0],
        [0,0,0,0,1,9,0,0,5],
        [0,0,0,0,0,0,0,0,0]]

def possible(row, column, number):
    global grid
    #Is the number appearing in the given row?
    for i in range(0,9):
        if grid[row][i] == number:
            return False

    #Is the number appearing in the given column?
    for i in range(0,9):
        if grid[i][column] == number:
            return False

    #Is the number appearing in the given square?
    x0 = (column // 3) * 3
    y0 = (row // 3) * 3
    for i in range(0,3):
        for j in range(0,3):
            if grid[y0+i][x0+j] == number:
                return False

    return True

def solve():
```

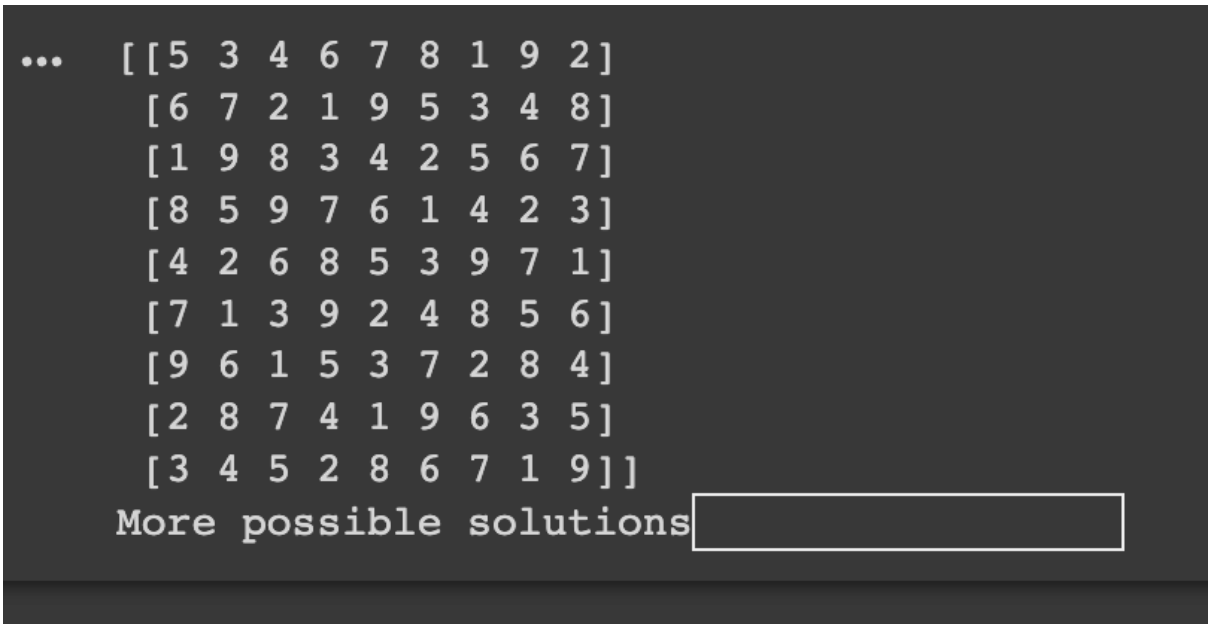
```
    global grid
    for row in range(0,9):
        for column in range(0,9):
            if grid[row][column] == 0:
                for number in range(1,10):
                    if possible(row, column, number):
                        grid[row][column] = number
                        solve()
                        grid[row][column] = 0

                return

    print(np.matrix(grid))
    input('More possible solutions')

solve()
```

OUTPUT:



```
[[5 3 4 6 7 8 1 9 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 9 7 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 7 1 9]]
More possible solutions
[[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]]
More possible solutions
```

If there is another solution for the given sudoku, The algorithm will find another solution too.

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

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

Dataset(s)

column,row,grid,number

```
grid =[[5,3,0,0,7,0,0,0,0],
       [6,0,0,1,9,5,0,0,0],
       [0,9,8,0,0,0,0,6,0],
       [8,0,0,0,6,0,0,0,3],
       [4,0,0,8,0,3,0,0,1],
       [7,0,0,0,2,0,0,0,6],
       [0,6,0,0,0,0,2,8,0],
       [0,0,0,0,1,9,0,0,5],
       [0,0,0,0,0,0,0,0,0]]
```

ALGORITHM

1. Create grid array that shows numbers in rows and columns.
2. If the number is already written in row, return false
3. If the number is already written in column, return false
4. If the number is already written in given square, return false
5. If none of the functions returns false, return true
6. If row,column is 0, then write 1-10 according to step 8-9-10

ASLI KOÇ 2104010063