

IT251 – DATA STRUCTURES & ALGORITHMS II

ASSIGNMENT 5

Name: **Sachin Prasanna**

Roll No.: **211IT058**

Problem Statement:

Graph Colouring Problem: Giant Sudoku

- Check the following literature on Internet about Graph coloring problem
 - <https://medium.com/code-science/sudoku-solver-graph-coloring-8f1b4df47072>
- Submission requirements
 - Write an intro on your understanding of Graph coloring problem and the proposed solution methodology (as per the write up above) – Not exceeding one page **(3 Marks)**
 - Understand the solution proposed about building a sudoku solver program
 - Design a build an “inspired solution” to extend the sudoku solver to solve any given sudoku with the following requirements **(10 Marks)**
 - Total number of cells in the grid – 256
 - Number of rows = 16; Number of columns = 16
 - Fill the empty cells in such a way that each row and column and region (4x4) has the following alpha numerals appearing only once
 - 1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G
 - Write down the relevant test cases to check the allowed conditions and illegal placements of numbers **(4 Marks)**
 - Provide a link or submit the code of your implementation clearly explaining the various APIs, functions in the code **(2 Marks)**
 - Attach an output to prove that your program works **(1 Mark)**

Note: Optionally, you can either build an interface that can take inputs from the user towards the empty cells OR you can use any built-in library that will create a sudoku puzzle for you (you can find them on internet for the language that you are using)

Tip: It will be lot easier to program this solution in Python since the example program in the link above is python based.

Answer:

Flow of the program

The program starts with a user input, where the user is presented with 2 options,

- 1 – Generate a random sudoku and solve it
- 2 – Generate an erroneous sudoku and show that it cannot be solved.

Any other digit entered, will cause the program to exit as supported by the output screenshot below:

```
===== RESTART: C:\Users\91900\Desktop\Assignment 5\main.py =====
***WELCOME TO 16x16 RANDOM SUDOKU SOLVER***
PRESS
1 - Generate a Valid Sudoku and Solve it
2 - Generate an erraneous Sudoku and show that it cannot be solved

5

INVALID VALUE ENTERED, BYE!
```

Sudoku Generation

The 16x16 sudokus are generated by the **Python Sudoku** module. The Python Sudoku module is a package that provides functions and classes for generating, solving, and validating Sudoku puzzles.

The Sudoku module provides a function called "generate" that can generate Sudoku puzzles of varying difficulties. User can specify the difficulty level and the number of puzzles to generate as arguments to this function. In our code, we are generating a 16x16 sudoku puzzle and then passing to our graph colouring algorithm to solve the puzzle.

Test Case 1 (Screenshots of output):

```
===== RESTART: C:\Users\91900\Desktop\Assignment 5\main.py =====
***WELCOME TO 16x16 RANDOM SUDOKU SOLVER***
PRESS
1 - Generate a Valid Sudoku and Solve it
2 - Generate an erroneous Sudoku and show that it cannot be solved

1

THE RANDOM SUDOKU GENERATED IS:

- - - - -
| 3 7 5 9 | B 0 G 0 | 1 2 4 6 | 0 0 D 0 |
| 2 1 F G | A C D 0 | 7 8 B 0 | 3 4 5 0 |
| A B C 4 | 6 0 0 7 | 3 0 D E | 0 G 8 9 |
| 6 8 D E | 3 0 5 9 | A F G C | 1 2 0 B |
- - - - -
| D 0 0 C | 8 9 0 0 | 0 3 A 7 | 6 5 0 G |
| E 0 9 6 | 4 3 B F | D G 1 5 | 7 8 C A |
| G A 8 F | 7 0 6 C | 9 B E 4 | 2 1 0 D |
| 5 3 7 0 | D A 1 G | 8 0 C 2 | E 9 0 4 |
- - - - -
| 4 C 6 0 | 9 8 0 1 | E 7 5 0 | D B A F |
| 1 D A 5 | G 7 C 4 | B 9 0 F | 8 6 E 2 |
| 9 0 E 7 | 5 0 A 6 | 4 0 2 8 | C 0 0 1 |
| B G 3 8 | 2 0 F D | 0 1 6 A | 0 7 9 0 |
- - - - -
| 7 5 B 3 | 1 6 0 A | 2 E 9 0 | 0 F 0 C |
| 0 6 2 A | F D 9 5 | G 4 7 3 | B E 0 8 |
| 8 9 4 D | E G 7 B | 6 0 0 1 | 5 A 2 3 |
| F E 0 0 | C 2 4 3 | 5 A 8 B | 0 D 0 7 |
- - - - -

SOLVING ...
```

SOLUTION IS:

	3	7	5	9		B	F	G	8		1	2	4	6		A	C	D	E	
	2	1	F	G		A	C	D	E		7	8	B	9		3	4	5	6	
	A	B	C	4		6	1	2	7		3	5	D	E		F	G	8	9	
	6	8	D	E		3	4	5	9		A	F	G	C		1	2	7	B	
	D	4	1	C		8	9	E	2		F	3	A	7		6	5	B	G	
	E	2	9	6		4	3	B	F		D	G	1	5		7	8	C	A	
	G	A	8	F		7	5	6	C		9	B	E	4		2	1	3	D	
	5	3	7	B		D	A	1	G		8	6	C	2		E	9	F	4	
	4	C	6	2		9	8	3	1		E	7	5	G		D	B	A	F	
	1	D	A	5		G	7	C	4		B	9	3	F		8	6	E	2	
	9	F	E	7		5	B	A	6		4	D	2	8		C	3	G	1	
	B	G	3	8		2	E	F	D		C	1	6	A		4	7	9	5	
	7	5	B	3		1	6	8	A		2	E	9	D		G	F	4	C	
	C	6	2	A		F	D	9	5		G	4	7	3		B	E	1	8	
	8	9	4	D		E	G	7	B		6	C	F	1		5	A	2	3	
	F	E	G	1		C	2	4	3		5	A	8	B		9	D	6	7	

PTO

Test Case 2 (Screenshots of output):

```
===== RESTART: C:\Users\91900\Desktop\Assignment 5\main.py =====
***WELCOME TO 16x16 RANDOM SUDOKU SOLVER***
PRESS
1 - Generate a Valid Sudoku and Solve it
2 - Generate an erroneous Sudoku and show that it cannot be solved

1

THE RANDOM SUDOKU GENERATED IS:

- - - - -
| C D 3 E | 1 9 A 2 | F G 7 8 | 4 0 5 6 |
| 7 A B 0 | 3 5 8 4 | 1 2 6 9 | C E 0 D |
| 2 6 0 G | B C 7 D | 4 3 5 E | 9 0 1 F |
| 4 1 5 9 | E 6 F G | C 0 D A | 0 3 7 0 |
- - - - -
| 3 7 C 6 | 2 E 9 F | 0 1 4 G | 0 8 A B |
| 0 2 F 5 | 0 0 D B | A 6 0 3 | 1 7 4 9 |
| 8 4 1 B | 6 3 5 0 | 7 0 9 0 | F 0 E G |
| 0 9 G A | 4 8 1 7 | B F E C | 5 2 6 3 |
- - - - -
| 0 C A 3 | 5 0 G 6 | D 0 2 B | E 9 8 1 |
| 1 G 0 0 | D F 0 9 | E 5 3 6 | 0 0 C 0 |
| 5 B 4 D | 8 0 C E | 9 7 A 0 | G 6 3 2 |
| 6 E 0 7 | A 0 4 3 | G 8 C 1 | B D F 5 |
- - - - -
| 0 8 0 4 | G B 2 C | 0 9 1 5 | 6 F D E |
| 0 0 D 2 | F A 0 5 | 6 0 B 4 | 8 0 0 7 |
| 0 5 E C | 0 0 6 1 | 8 A F 7 | 3 0 2 0 |
| 0 F 6 1 | 7 4 0 8 | 2 0 G 0 | 0 5 B C |
- - - - -

SOLVING ...
```

PTO

SOLUTION IS:

C	D	3	E	1	9	A	2	F	G	7	8	4	B	5	6
7	A	B	F	3	5	8	4	1	2	6	9	C	E	G	D
2	6	8	G	B	C	7	D	4	3	5	E	9	A	1	F
4	1	5	9	E	6	F	G	C	B	D	A	2	3	7	8
3	7	C	6	2	E	9	F	5	1	4	G	D	8	A	B
E	2	F	5	C	G	D	B	A	6	8	3	1	7	4	9
8	4	1	B	6	3	5	A	7	D	9	2	F	C	E	G
D	9	G	A	4	8	1	7	B	F	E	C	5	2	6	3
F	C	A	3	5	7	G	6	D	4	2	B	E	9	8	1
1	G	2	8	D	F	B	9	E	5	3	6	7	4	C	A
5	B	4	D	8	1	C	E	9	7	A	F	G	6	3	2
6	E	9	7	A	2	4	3	G	8	C	1	B	D	F	5
A	8	7	4	G	B	2	C	3	9	1	5	6	F	D	E
G	3	D	2	F	A	E	5	6	C	B	4	8	1	9	7
B	5	E	C	9	D	6	1	8	A	F	7	3	G	2	4
9	F	6	1	7	4	3	8	2	E	G	D	A	5	B	C

Error Handling

To generate erroneous test cases, a loop is run and a random integer between 1 to 9 is selected. Then, this random integer is placed side by side in all the rows. This clearly means the sudoku is **illegal** and it cannot be solved. The outputs in case of **illegal** Cases are as follows:

Test Case 3 (Screenshots of output):

In this case, the random integer generated was 9. Observing the generated sudoku below, it can easily be observed that 9 is present twice in all rows, hence making the sudoku unsolvable.

When this sudoku is given to the algorithm to be solved, it simply outputs **NO SOLUTION EXISTS** as it rightly should.

WELCOME TO THE RANDOM SUDOKU SOLVER
PRESS
1 - Generate a Valid Sudoku and Solve it
2 - Generate an erroneous Sudoku and show that it cannot be solved

2

THE RANDOM SUDOKU GENERATED IS:

```

- - - - -
| 9 9 B 0 | 0 6 E F | 5 1 0 G | 3 7 9 A |
| 2 9 9 1 | 4 0 3 A | 9 8 6 B | C G D 0 |
| D 0 9 9 | B C G 0 | A 0 E 0 | 1 2 4 5 |
| 3 A F 9 | 9 5 9 D | 2 4 7 0 | 8 6 B E |
- - - - -
| 4 E 2 7 | 9 9 B 6 | F A G 0 | 5 0 1 0 |
| 9 C 5 0 | F 9 9 0 | 8 B 1 E | D 0 6 7 |
| 1 G 0 0 | 8 E 9 9 | 7 9 5 0 | F A C 2 |
| A D 8 F | 0 1 5 9 | 9 0 0 6 | B 3 E 0 |
- - - - -
| E 3 1 0 | D 0 7 0 | 9 9 4 2 | 0 0 A C |
| B 7 4 6 | G 0 2 C | 3 9 9 5 | E 1 0 9 |
| F 0 A 0 | 5 4 6 E | C 7 9 9 | G 0 3 D |
| G 9 C 0 | 3 A 0 1 | 6 E B 9 | 9 5 7 4 |
- - - - -
| 6 B 3 C | 9 2 8 4 | E 0 0 7 | 9 9 5 0 |
| 7 2 D E | A 3 C 5 | 1 6 8 0 | 4 9 9 0 |
| 8 F G 4 | E B 1 7 | D 5 3 A | 0 C 9 9 |
| 5 0 0 A | 6 F D G | B 2 0 4 | 7 E 0 3 |
- - - - -

```

SOLVING ...

SORRY, NO SOLUTION EXISTS!

PTO

Test Case 4 (Screenshots of output):

```
WELCOME TO THIS RANDOM SUDOKU SOLVER
PRESS
1 - Generate a Valid Sudoku and Solve it
2 - Generate an erroneous Sudoku and show that it cannot be solved

2

THE RANDOM SUDOKU GENERATED IS:

- - - - -
| 1 1 4 1 | E A 8 0 | G F B 5 | 2 3 D 7 |
| 7 1 1 2 | 0 G 0 C | 3 A D 4 | 0 1 6 E |
| E 0 1 1 | 1 D 2 4 | 8 C 0 0 | G 9 F 0 |
| G F 0 1 | 1 0 0 7 | 2 0 9 0 | 8 A 4 C |
- - - - -
| 0 C 0 6 | 1 1 D E | 1 0 A 0 | 7 5 2 8 |
| 1 D E 7 | 0 1 1 A | 0 2 3 8 | B C 0 9 |
| F 2 G 0 | 0 7 1 1 | B D E 0 | A 6 1 4 |
| 4 9 A 5 | 8 0 1 1 | 1 7 G 6 | D F E 3 |
- - - - -
| 3 7 8 A | 2 B C 0 | 1 1 F 0 | 1 4 5 6 |
| 2 B 0 E | 4 6 F 5 | 7 1 1 A | 9 G C D |
| 5 4 C 0 | 9 E 7 G | 6 B 1 1 | 3 0 8 A |
| D G 6 9 | 3 8 A 1 | 4 5 2 1 | 1 0 B F |
- - - - -
| 6 3 0 G | D 4 0 8 | F 9 7 1 | 1 1 0 5 |
| 8 5 7 0 | A 1 3 F | 0 6 0 2 | 0 1 1 G |
| 9 0 0 C | 7 5 G 6 | 0 8 0 B | F E 1 1 |
| A 0 F 4 | B C 9 0 | 0 G 5 3 | 0 8 0 1 |
- - - - -

SOLVING ...

SORRY, NO SOLUTION EXISTS!
```

Code Files:

The code files are documented and submitted in the same folder as well. They can be run on any computer which can run python and have the Sudoku module in them. If the sudoku module is not available, please type the following command on command prompt and then the run the code files:


```
# Python 2
```

```
pip install py-sudoku
```

```
# Python 3
```

```
pip3 install py-sudoku
```
