

Assignment of Combinatorics and Graphs

02/2020

1 Objective

In this Assignment you have to build Sudoku puzzles using permutation and output them to file.

2 Sudoku problem

2.1 Definitions

The puzzle consists of a 9×9 grid in which some of the entries of the grid have a number from 1 to 9. A traditional Sudoku puzzle 9 cells divided into 3×3 subsections called blocks. A Sudoku solution must satisfy the rules of Sudoku following:

- Numbers in rows are not repeated
- Numbers in columns are not repeated
- Numbers in 3×3 blocks are not repeated
- Order of the numbers when filling is not important

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

(a) The Puzzle

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

(b) The Solution

Pic. 1 – A sample of Sudoku puzzle and Solution

2.2 Build Sudoku puzzle

We start with a valid solution and generate a puzzle by picking some holes.

2.2.1 Latin Squares for Solution Generation

To quickly and effectively generate a solution, we use the 12 unique 3×3 Latin Squares.

Select nine 3×3 Latin Squares, with replacement.

Place each of these squares into one of the blocks in a blank grid. In this step, you must guarantee that do not any duplicated number on row and column matrix as figure below.

0	1	2
1	2	0
2	0	1

Select another 3 × 3 Latin Square and match each cell with the corresponding block in the Sudoku grid. The same as the previous step, you must guarantee that do not any duplicated block in all blocks except outside block as Fig (a).

Each cell now has a pair of numbers. Treat these pairs as base 3 numbers, and convert to base 10, adding 1 (Fig (b)). For example, with the block following:

0	2	1
1	0	2
2	1	0

You will obtain the first row as below:

$$2 \times 3 + 0 + 1 = 7$$

$$2 \times 3 + 2 + 1 = 9$$

$$2 \times 3 + 1 + 1 = 8$$

the second row:

$$2 \times 3 + 1 + 1 = 8$$

$$2 \times 3 + 0 + 1 = 7$$

$$2 \times 3 + 2 + 1 = 9$$

and the third row:

$$2 \times 3 + 2 + 1 = 9$$

$$2 \times 3 + 1 + 1 = 8$$

$$2 \times 3 + 0 + 1 = 7$$

Finally, you will have the result as following:

7	9	8
8	7	9
9	8	7

Each cell now has the numbers 1-9. However blocks contain duplicates. So that you will swap the 2nd & 4th rows, 3rd & 7th rows, and 6th & 8th rows preserving the row and column properties, and adding the desired property for blocks. Finally, we will obtain a valid solution as Fig (c)

0	2	1	1	0	2	0	1	2
1	0	2	0	2	1	1	2	0
2	1	0	2	1	0	2	0	1
2	0	1	2	1	0	2	1	0
1	2	0	1	0	2	0	2	1
0	1	2	0	2	1	1	0	2
0	1	2	2	0	1	0	2	1
2	0	1	0	1	2	2	1	0
1	2	0	1	2	0	1	0	2

(a) Selection of Latin Squares

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

(b) Conversion to Base 10

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

(c) Sudoku Solution

2.2.2 Digging hole

After generating solution for Sudoku puzzle, you will dig some holes randomly to make Sudoku puzzle. For example, you will obtain Sudoku puzzle as following with the number of holes is 50.

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

3 Assignment Requirements

Use programming language Python3 (the pure version without using any external packages like numpy, pandas, pytorch,...) to write your program. Your program is named as `assignment_mssv.py`, where `mssv` is replaced by your student ID and all characters are lowercased.

Your program is command-line based.

Users will give the number of holes as the first argument. The number of holes is multiple of 9 (for instance, 9, 18, 27, 36, 45, 54). The holes are equally distributed to all cells (for example, if the number of holes is 18, then each cell has 2 holes). Users will give the name of the output puzzle as the second argument.

Your program will output to file a puzzle containing holes. In the output file 0 represents holes

You have to import package *random* (built-in package in Python3) and apply *seed* function writing these commands in the beginning of your program

```
import random  
random.seed(m)
```

where m is replaced by your student ID taking only digits (for example, student with ID 517H0050 will use $m = 5170050$)

An example of running your program

```
$python3 assignment_mssv.py 9 output.txt
```

The content of output.txt file is as follows:

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

4 Submission

- Submit your assignment solution to the elit.
- First Deadline: April 23, 2020 (you will get the maximum score 10)
- Final Deadline: April 30, 2020 (you will get the maximum score 9)

5 References

1. Vo Hoang Anh, [2019], Combinatorics and Graphs assignment description.
2. Khee-Meng Koh, Chuan Chong Chen, [1992], Principles and Techniques in Combinatorics, World Scientific Publishing Company.
3. John Harris, Jeffry L. Hirst, Michael Mossinghoff, [2008], Combinatorics and Graph Theory (2nd Edition), Springer.
4. C. Vasudev, [2007], Combinatorics and Graph Theory, New Age International Pvt Ltd Publishers.
5. Miklos Bona, [2011], A Walk Through Combinatorics: An Introduction to Enumeration and Graph Theory (3rd Edition), World Scientific Publishing Company.
6. Alan Tucker, [2012], Applied Combinatorics (6th Edition), Wiley.
7. Sudoku: Bagging a Difficulty Metric & Building Up Puzzles