VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
INFORMATION TECHNOLOGY FACULTY

COMBINATORICS AND GRAPHS

GENERATE SUDOKU PUZZLE

Instructing Lecturer: Mr. NGUYEN CHI THIEN

Student's name: **HUỲNH LÊ THIÊN Ý – 518H0320**

Class **: 18H50204**

Course **: 22**

HO CHI MINH CITY, YEAR 2020

VIETNAM GENERAL CONFEDERATION OF LABOUR
TON DUC THANG UNIVERSITY
INFORMATION TECHNOLOGY FACULTY

COMBINATORICS AND GRAPHS

GENERATE SUDOKU PUZZLE

Instructing Lecturer: Mr. NGUYEN CHI THIEN

Student's name: **HUỲNH LÊ THIÊN Ý – 518H0320**

Class **: 18H50204**

Course **: 22**

HO CHI MINH CITY, YEAR 2020

ACKNOWLEDGEMENT

In order to make this report complete and achieve good results, we have received the support and assistance of many teachers and classmates. With deep affection, sincerity, we express deep gratitude to all individuals and agencies who have helped us in our study and research. First of all, I would like to express a special appreciation to Ton Duc Thang.

University's teachers for their conscientious guidance and advices throughout the last semester by gave me their modern outlook and meticulous supervision to carry out the job perfectlyWe would like to express our sincere gratitude to the leadership of Ton Duc Thang University for supporting, helping and facilitating us to complete the report well during the study period. With limited time and experience, this report cannot avoid mistakes. We are looking forward to receiving advice and comments from teachers so that we can improve our awareness, better serve the practical work later. We sincerely thank you!

THE PROJECT WAS COMPLETED AT TON DUC THANG UNVERSITY

The content of research, results in this subject is honest and not published in any form before. The data in the tables used for the analysis, comment, and evaluation were collected by the authors themselves from various sources indicated in the reference section. In addition, many comments and assessments as well as data from other authors and organizations have been used in the project, with references and annotations. If any fraud is found, I am fully responsible for the content of my project. Ton Duc Thang University is not involved in any copyright infringement or copyright infringement in the course of implementation (if any).

Ho Chi Minh City, April 19 2020

Author

Signed

Huỳnh Lê Thiên Ý

EVALUATION OF INSTRUCTING LECTURER

Confirmation of the instructor

_____
_____
_____
_____
_____
_____
_____
_____
_____

Ho Chi Minh City, month day 2020

(Sign and write full name)

The assessment of the teacher marked

_____
_____
_____
_____
_____
_____

Ho Chi Minh City, month day 2020

(Sign and write full name)

TABLE OF CONTENTS

LIST OF TABLES, DRAWINGS, GRAPHICS

INTRODUCTION

Sudoku is a logic puzzle, originally coming from Japan. Within the Western world, it's caught on in popularity enormously over the last few years. Most newspapers now publish a Sudoku puzzle for the readers to unravel daily.

A Sudoku puzzle consists of a 9x9 grid. A number of the cells within the grid have digits (from 1 to 9), others are blank. Here is an example of a Sudoku puzzle:

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

Figure 1 An example of a Sudoku puzzle

Following are the rules of Sudoku for a player.

- In all 9 sub matrices 3×3 the elements should be 1 to 9 and not repeated.
- In all rows there should be elements between 1 to 9 and not repeated.
- In all columns there should be elements between 1 to 9 and not repeated.

The program generate a 9 x 9 Suduku grid that is valid, i.e., a player can fill the grid following above set of rules.

ALGORITHM DESCRIPTION

The Sudoku Generator algorithm need to use a backtracking algorithm to check if a grid which be generated is solvable and check that it only has one solution.

The backtracking algorithm is used to check all possible solutions of a given grid. It is a recursive algorithm that attempts to solve a problem by checking all possible ways until a solution is found. The Sudoku Generator algorithm is predicated on 5 steps:

1. Generate a full grid of numbers (fully filled in). This step is more complex because it seems as it cannot just randomly generate numbers to fill within the grid. These numbers must be positioned on the grid following the Sudoku rules. To complete the puzzle the program use a backtracking algorithm that will apply to an empty grid. Add a random element to this backtracking algorithm to make sure that a replacement grid is generated each time I run it.
2. From the full grid, the program will remove one number at a time.
3. Each time a number is removed, it will apply a backtracking algorithm to determine if the grid can still be solved and count the amount of solution its have.
4. If the resulting grid only has one solution, then the program is able to stick with it the process from step 2. If not the program must put the number which been taken away back in the grid.
5. Repeat the same process (from step 2) several times using a different number each time to try to get rid of additional numbers, leading to a more difficult grid to solve. The amount of attempts will use to travel through this process will have an impact on the difficulty level of the resulting grid.

RESULT OF PROGRAM

My program is command-line based and in program language Python 3.

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.

0, 5, 6, 4, 2, 1, 0, 3, 7

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

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

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

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

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

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

4, 0, 7, 0, 6, 2, 3, 5, 1

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

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

Figure 2 python assignment_518H0320.py 9 out.txt

0, 5, 6, 4, 2, 1, 9, 0, 7

1, 7, 0, 0, 9, 3, 5, 4, 8

3, 4, 9, 5, 8, 0, 0, 1, 6

7, 6, 1, 2, 5, 0, 0, 8, 3

0, 0, 4, 1, 3, 0, 7, 2, 5

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

0, 3, 0, 9, 0, 4, 0, 7, 2

4, 9, 7, 8, 6, 0, 3, 0, 1

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

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

Figure 3 python assignment_518H0320.py 18 out.txt

0, 5, 6, 4, 0, 0, 0, 3, 7

1, 0, 0, 6, 9, 0, 5, 4, 0

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

0, 6, 0, 0, 0, 9, 4, 8, 3

0, 8, 4, 1, 0, 6, 0, 2, 0

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

6, 3, 5, 0, 0, 4, 8, 7, 0

0, 9, 7, 0, 6, 2, 3, 5, 0

0, 1, 0, 3, 7, 5, 0, 9, 4

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

Figure 4 python assignment_518H0320.py 27 out.txt

0, 5, 6, 0, 0, 1, 0, 3, 7

0, 0, 0, 6, 9, 0, 5, 0, 0

3, 4, 9, 5, 8, 0, 0, 1, 6

0, 6, 0, 0, 0, 0, 4, 0, 3

0, 8, 4, 0, 3, 6, 7, 2, 0

5, 2, 0, 7, 4, 8, 0, 0, 9

6, 0, 0, 0, 1, 0, 8, 0, 0

4, 9, 0, 0, 6, 2, 3, 5, 1

0, 1, 8, 3, 7, 0, 0, 9, 0

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

Figure 5 python assignment_518H0320.py 36 out.txt

0, 5, 0, 0, 0, 1, 9, 0, 0

0, 0, 0, 0, 9, 0, 5, 4, 0

3, 4, 9, 0, 8, 7, 0, 0, 6

7, 0, 0, 0, 0, 0, 4, 0, 0

0, 8, 4, 0, 3, 6, 7, 2, 5

0, 2, 0, 7, 4, 0, 0, 0, 0

0, 0, 5, 0, 1, 4, 8, 7, 0

4, 0, 0, 0, 0, 0, 3, 5, 0

2, 1, 0, 3, 7, 0, 0, 0, 0

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

Figure 6 python assignment_518H0320.py 45 out.txt

0, 5, 0, 0, 0, 0, 0, 0, 7

0, 0, 0, 0, 9, 0, 0, 4, 8

3, 4, 0, 0, 8, 7, 0, 0, 0

7, 0, 0, 2, 0, 0, 0, 8, 0

0, 8, 0, 1, 3, 0, 0, 0, 0

0, 2, 0, 0, 0, 0, 1, 6, 0

0, 0, 5, 9, 0, 0, 0, 0, 0

0, 0, 0, 8, 6, 0, 0, 5, 0

0, 1, 8, 0, 0, 0, 0, 9, 4

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

Figure 7 python assignment_518H0320.py 54 out.txt

CONCLUSION

Achievements

Through this assignments I've got improved my programming ability. The code isn't in highest quality due to severely lacking in documentation, but the assignment requirement were properly completed. Finally, I've got experienced participating in what can be called a small scientific research (useful for future work).

Project Status

The logic of the program is sufficient for generate many Sudoku puzzles. However, the implementation could likely be improved to execute faster. Furthermore, there are many logic solving techniques that haven't been implemented. The generator, on the opposite hand, is currently rather poorly implemented. It works in this it successfully generates grids, but the variability of the time taken to come up with could be a significant we akness.

Suggest for Future Improvement

These are some of the main suggestions given by the user for improvement:

1. Add user interface and make it into a game.
2. Offers levels when generate sudoku by giving more attempts.
3. Add score system base on time and accuracy.

REFERENCES

[1] https://www.ocf.berkeley.edu/~arel/sudoku/main.html
[2] https://www.101computing.net/sudoku-generator-algorithm/
[3] https://www.101computing.net/backtracking-algorithm-sudoku-solver/
[4] https://github.com/mld2443/Sudoku
[5] https://www.geeksforgeeks.org/sudoku-backtracking-7/

SELF EVALUATION

Criterion	Marking scheme	1	2	3	Self-evaluated	Reason
	Mark on 10	0 mark	½ mark	Full mark		
1/ Form	**4.0**					
Report (.docx)	3.0	Not report or not enough 50% all section	Report 50% all section	Report all section in highly quality	3.0	
Output in right required format	1.0	Output not in right format		Output in right format	1.0	
2/ Content	**4.0**					
Create all Latin Squares cells	3.0	Not create Latin Squares cells (evaluate by output)		Create Latin Squares cells (evaluate by output)	3.0	
Implemented Digging Holes	1.0	Not implemented Digging Holes (evaluate by output)		Implemented Digging Holes (evaluate by output)	1.0	
3/ Behavior	**2.0**					
Submit in time	2.0	Before 16.04.2020	Before 06.04.2020	Before 30.03.2020	2.0	
Total	**10**			**Result:**	**10**	