

SUDOKU SOLVER

Project Report

Subject: IT-208 Algorithm Design and Analysis

**BACHELOR OF TECHNOLOGY IN
INFORMATION TECHNOLOGY**

4th Semester

Submitted by: VARUN KUMAR (2K19/IT/140)

YASHIT KUMAR (2K19/IT/149)

Under the Supervision of

Ms. Ritu Agarwal

Assistant Professor



DEPARTMENT OF INFORMATION TECHNOLOGY DELHI TECHNOLOGICAL
UNIVERSITY (Formerly Delhi College of Engineering) Bawana Road, Delhi-110042

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

We Varun Kumar, Roll No – 2K19/IT/140 & Yashit Kumar, Roll No - 2K19/IT/149, students of B.Tech. (INFORMATION TECHNOLOGY), hereby declare that the project Dissertation titled “Sudoku Solver” which is submitted by us to the Department of INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the 4th semester of the Bachelor of Technology, is made by us. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi

Date: 17-04-2021

Varun Kumar

Yashit Kumar

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project: "**Sudoku Solver**" which is submitted by Varun Kumar, Roll No – 2K19/IT/140 & Yashit Kumar, Roll No – 2K19/IT/149, INFORMATION TECHNOLOGY, Delhi Technological University, Delhi in fulfillment of the requirement for the 4th semester of Bachelor of Technology, is record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Supervisor : Ms.Ritu Agarwal

Date: 17-April-2021

(Assistant Professor)

DEPARTMENT OF INFORMATION TECHNOLOGY

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We would like to convey our heartfelt thanks to our supervisor Ms.Ritu Agarwal for his ingenious ideas, tremendous help and cooperation.

We are extremely grateful to our friends who gave valuable suggestions and guidance for completion of our project. The cooperation and healthy criticism came handy and useful with them.

Finally we would like to thank all the above mentioned people once again.

TABLE OF CONTENTS

1 . INTRODUCTION	6 - 9
2. ALGORITHM	10 - 11
3. FEATURES	12
4. WORKING DEMO	13 – 22
5. BIBLIOGRAPHY	23

INTRODUCTION

The Sudoku puzzle has become a very popular puzzle.

The puzzle consists of a 9×9 grid in which some of the entries of the grid have a number from 1 to 9.

Filling the table with the numbers must follow these rules:--

- 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

SAMPLE SUDOKU PUZZLE

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

SOLVED SUDOKU PUZZLE

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

WHY SUDOKU ?

→ Keeps your brain alive and active, a useful break from your usual routine.

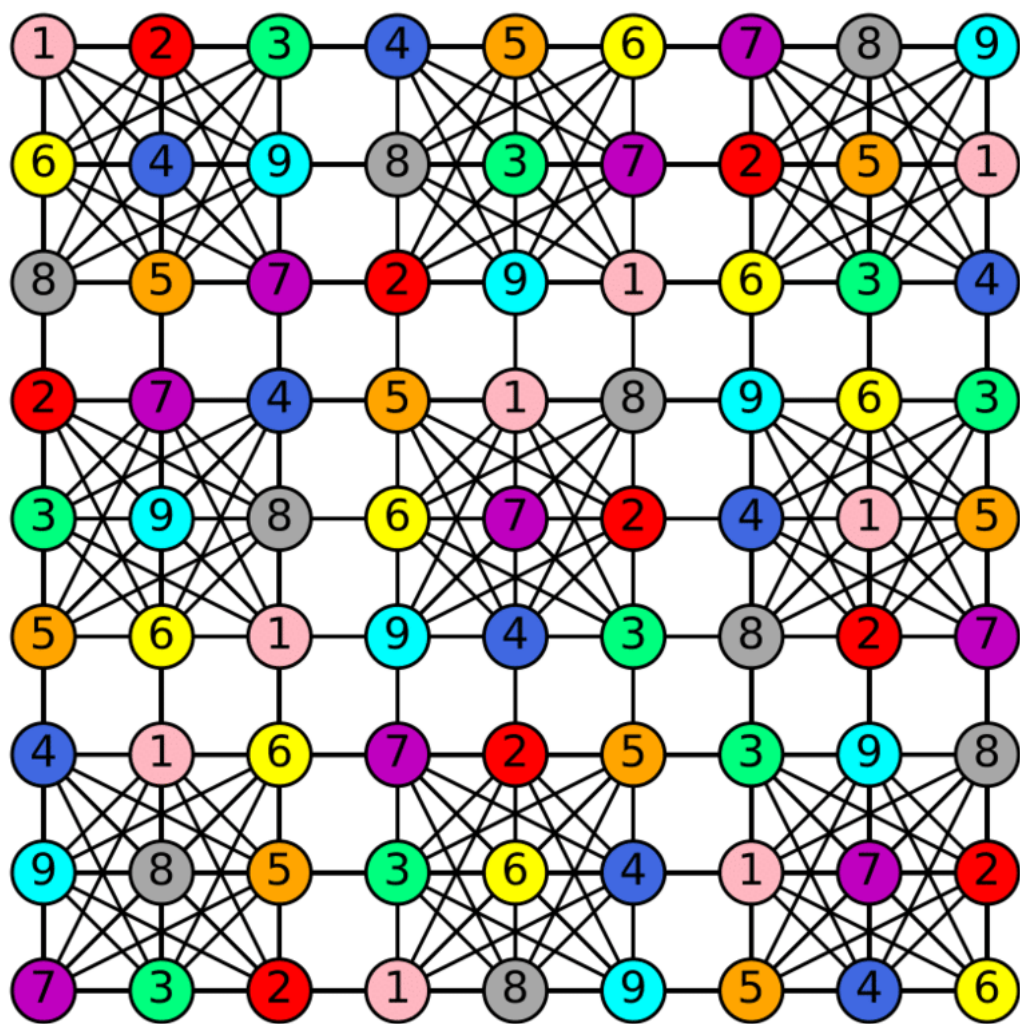
→ Develops some logical thinking and patterns.

→ Helps in improving your patience and concentration.

→ Improves sharpness and strategy while approaching and solving problems since you need to be accurate while putting digits and aware of the placement of other digits.

→ Gives you a sense of accomplishment on completing the grid within few minutes.

→ It is one of the simplest sports/games/activities to learn, enjoy, play/solve, and even teach.



ALGORITHM

➤ Recursion and Backtracking

- The idea is to assign numbers from 1 to 9 one by one to boxes, starting from first box .
- Before assigning a number , check for safety by considering already assigned number to the adjacent boxes.
- Check if the adjacent boxes have the same color or not by traversing the sudoku in the same row , column and subgrid.
- If the box we reached by traversing has the same number as of the initial box then we have to assign different number to the initial box .

- And if we are not able to assign any number to current box then we have to backtrack and change the number of previous box.
- If we are able to assign number to the box then we will move to next box in the row.
- If no assignment of number is possible then in that case we are not able to solve Sudoku and hence puzzle is unsolvable.

FEATURES

- User can play the game and It can be played in Easy , Medium and Hard mode
- User can solve its manually entered puzzle .
- User can get the hint while solving the puzzle
- User can erase its move.
- User can unlock the row or column while playing.
- User can get the complete solution if he wants.

Language Used : C++

WORKING DEMO

WELCOME TO
SUDOKU SOLVER
PROJECT

Created By:-

VARUN KUMAR
YASHIT KUMAR

LOADING 50%_

MAIN MENU

```
=====
#####:SUDOKU SOLVER !!!:#####
=====
```

```
-----
|          1. PLAY THE GAME          |
|-----
```

```
-----
|          2. ENTER PUZZLE TO SOLVE   |
|-----
```

```
-----
|          3. INSTRUCTIONS             |
|-----
```

```
-----
|          4. EXIT                    |
|-----
```

Enter your choice:->

PLAY GAME MENU

```
=====
#####:PLAY  GAME !!!:#####
=====
```

```
-----
|          1. EASY          |
|-----|
```

```
-----
|          2. MEDIUM       |
|-----|
```

```
-----
|          3. HARD         |
|-----|
```

```
-----
|          4. BACK TO MAIN MENU          |
|-----|
```

Enter your choice:->

ENTER PUZZLE MENU

```
#####: :PUZZLE !!!: :#####
```

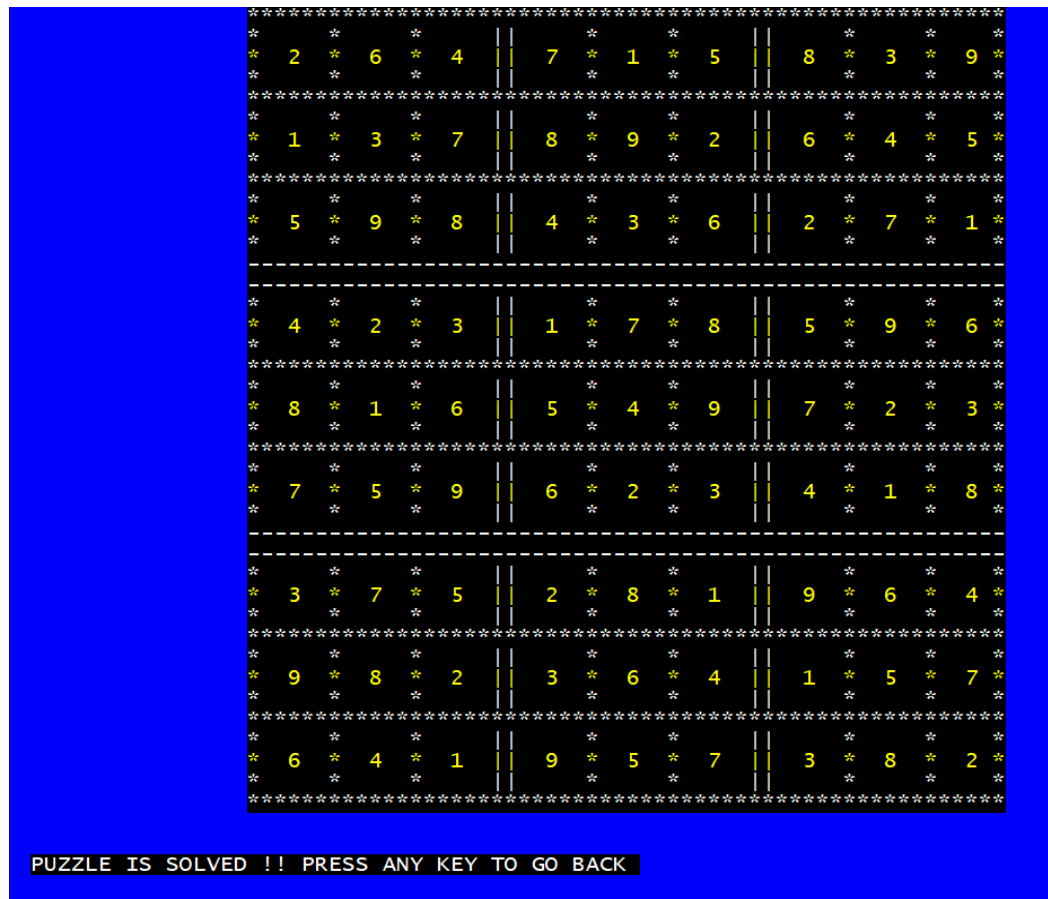
```
| 1. ENTER PUZZLE BY ROW COLUMN |
```

```
| 2. ENTER PUZZLE IN ONE LINE |
```

```
| 3. BACK TO MAIN MENU |
```

Enter your choice:->

USER CAN USE THESE FEATURES WHILE PLAYING THE GAME



ENTER O TO UNLOCK ROW!!!!!

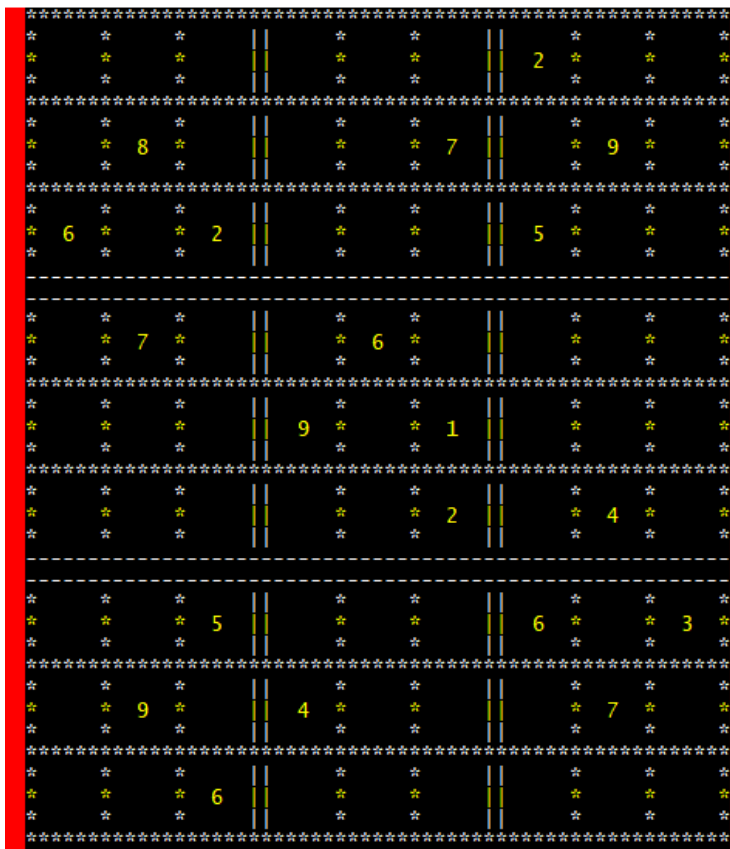
ENTER P TO UNLOCK COLUMN

ENTER H TO GET HINT!!!!!

ENTER C TO GET COMPLETE SOLUTION!!!!

ENTER E TO ERASE THE MOVE!!!!!

ENTER A TO EXIT!!!!!



ENTER O TO UNLOCK ROW!!!!

ENTER P TO UNLOCK COLUMN

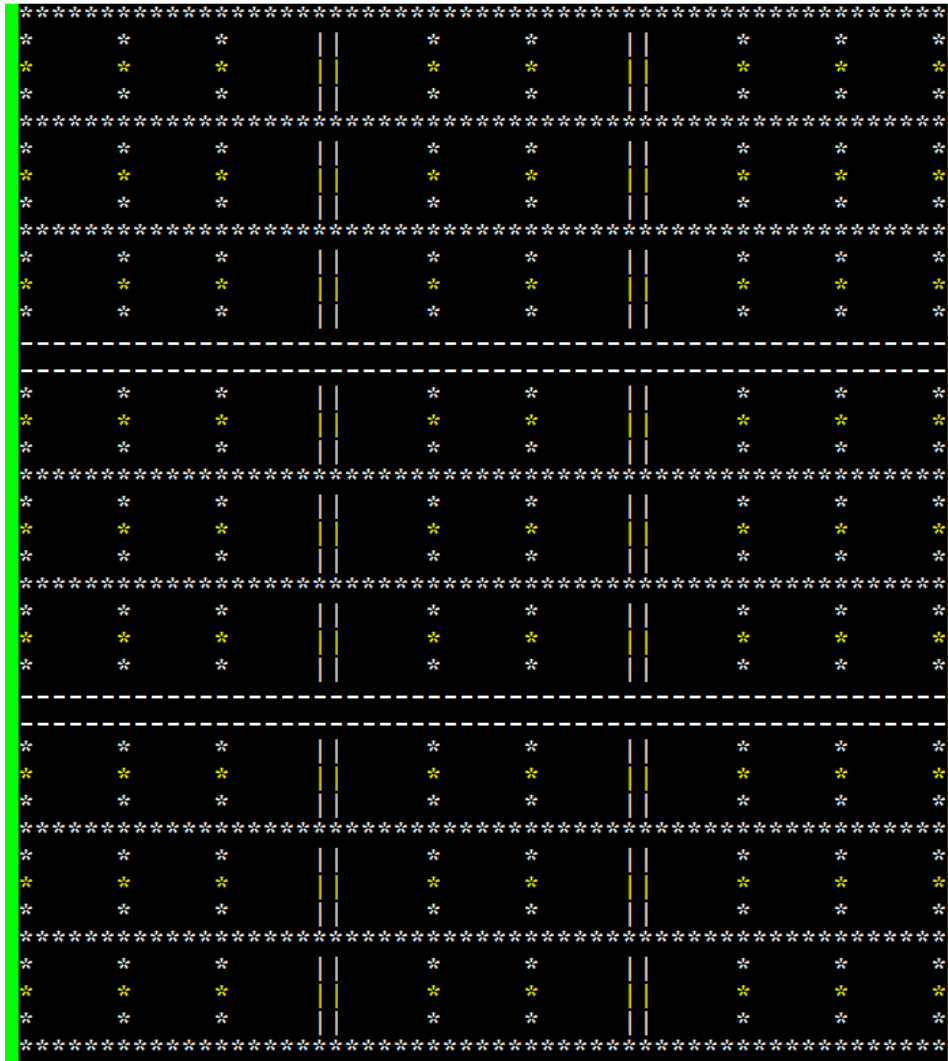
ENTER H TO GET HINT!!!!

ENTER C TO GET COMPLETE SOLUTION!!!!

ENTER E TO ERASE THE MOVE!!!!

ENTER A TO EXIT!!!!

USER CAN USE ENTER THE PUZZLE MANUALLY AND PLAY IT



ENTER E TO ERASE THE MOVE

ENTER S TO STOP ENTERING!!!!

ENTER A TO EXIT!!!!

*	*	*	*	*	*	*	*	*
*	2	*	6	*	4	7	*	1
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	1	*	3	*	7	8	*	9
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	5	*	9	*	8	4	*	3
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	4	*	2	*	3	1	*	7
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	8	*	1	*	6	5	*	4
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	7	*	5	*	9	6	*	2
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	3	*	7	*	5	2	*	8
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	9	*	8	*	2	3	*	6
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*

*	*	*	*	*	*	*	*	*
*	6	*	4	*	1	9	*	5
*	*	*	*	*	*</			

BIBLIOGRAPHY

https://en.wikipedia.org/wiki/Sudoku_graph

<https://www.geeksforgeeks.org/sudoku-backtracking-7/>

<https://www.geeksforgeeks.org/backtracking-introduction/>

<https://www.geeksforgeeks.org/recursion/>