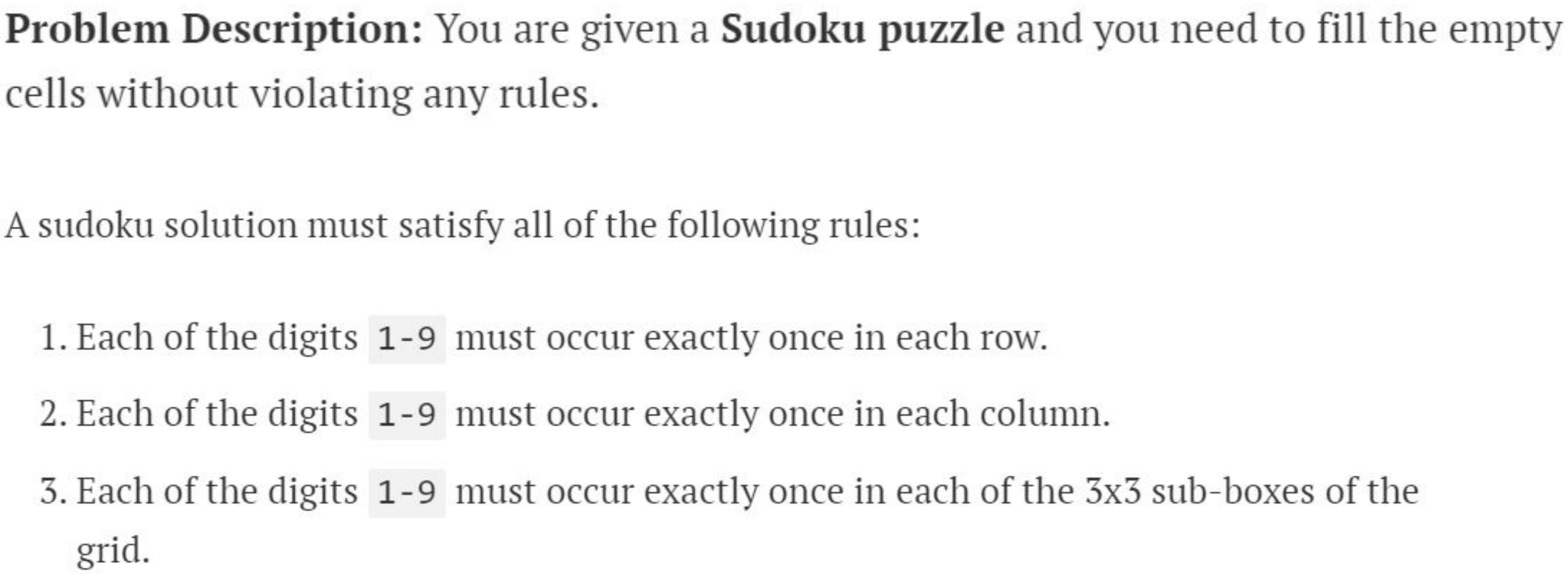
SUDOKU SOLVER

U19CS011 RAJ JIKADRA
U19CS012 BHAGYA RANA
U19CS049 DEV JARIWALA
U19CS080 MUKTESHARYAN UPPALA

SUDOKU SOLVER



Problem Description: You are given a **Sudoku puzzle** and you need to fill the empty cells without violating any rules.

A sudoku solution must satisfy all of the following rules:

1. Each of the digits `1-9` must occur exactly once in each row.
2. Each of the digits `1-9` must occur exactly once in each column.
3. Each of the digits `1-9` must occur exactly once in each of the 3x3 sub-boxes of the grid.

WHAT IS SUDOKU?

Sudoku is a logic-based, combinatorial number placement puzzle game where the objective is to fill a square grid of size ’n’ with numbers between 1 to ’n’.

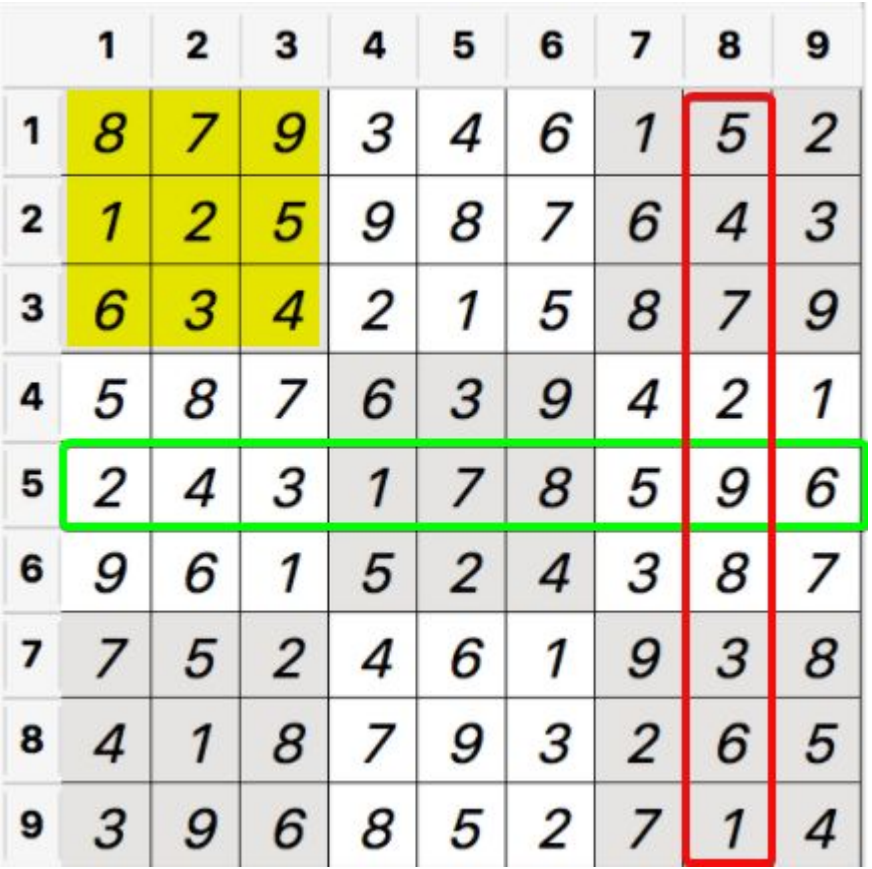
The numbers must be placed so that :

- each **column**,
- each **row**, and
- each of the **sub-grids** (if any) contains all of the numbers from **1 to ‘n’**.

EXAMPLE:

Standard Sudoku

- Grid size: 9×9
- Number of inner boxes: 9
- Inner box shape: 3×3 Square
- Number range: 1 to 9



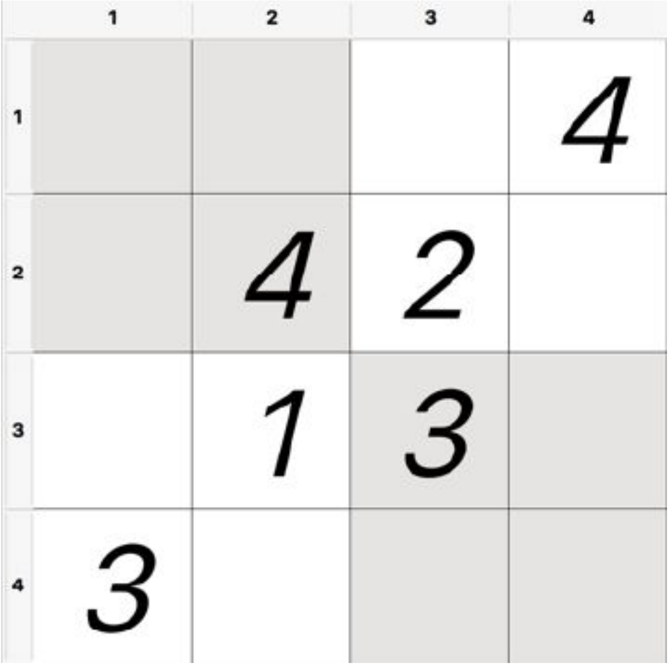
	1	2	3	4	5	6	7	8	9
1	8	7	9	3	4	6	1	5	2
2	1	2	5	9	8	7	6	4	3
3	6	3	4	2	1	5	8	7	9
4	5	8	7	6	3	9	4	2	1
5	2	4	3	1	7	8	5	9	6
6	9	6	1	5	2	4	3	8	7
7	7	5	2	4	6	1	9	3	8
8	4	1	8	7	9	3	2	6	5
9	3	9	6	8	5	2	7	1	4

4x4 SUDOKU?

Small Sudoku

Small Sudoku is probably the smallest and easiest Sudoku variant.

- Grid size: 4 × 4
- Number of inner boxes: 4
- Inner box shape: Square
- Number range: 1 to 4



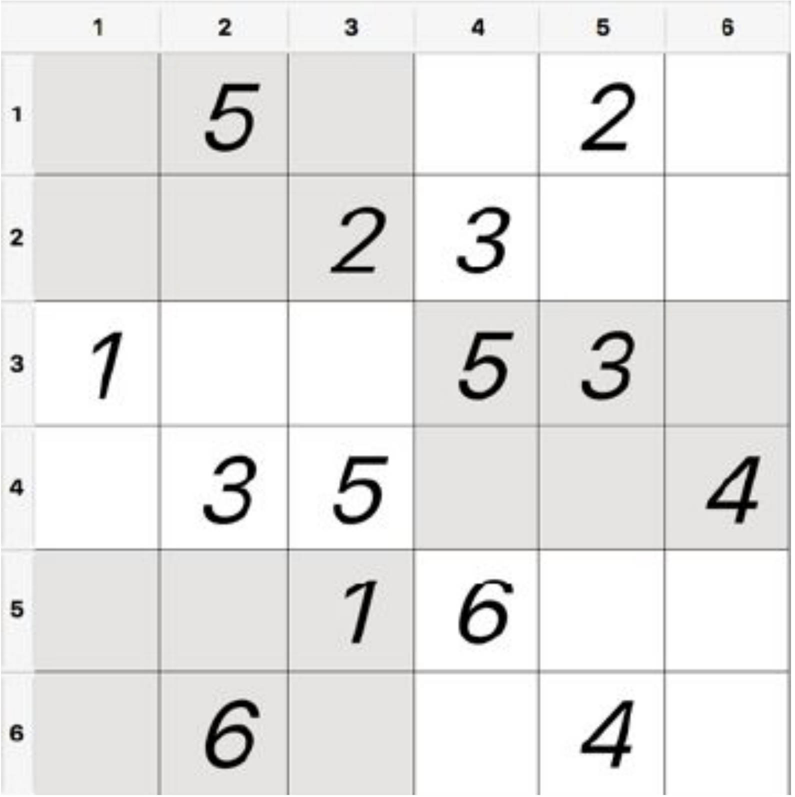
	1	2	3	4
1				4
2		4	2	
3		1	3	
4	3			

6x6 SUDOKU?

Mini Sudoku

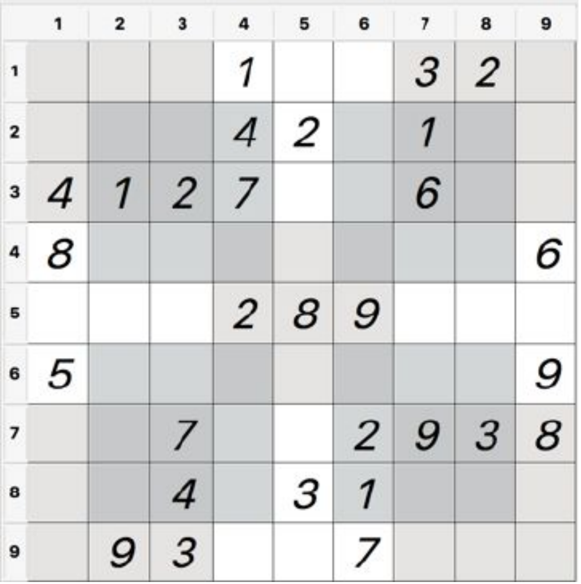
Mini Sudoku is a relatively easy variant with non-square inner boxes.

- Grid size: 6 × 6
- Number of inner boxes: 6
- Inner box shape: 3 × 2 Rectangle
- Number range: 1 to 6

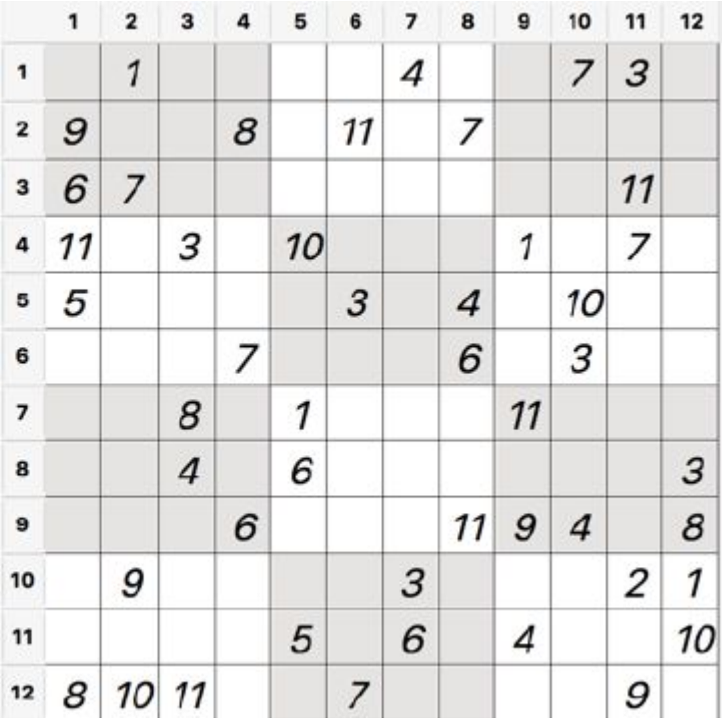


	1	2	3	4	5	6
1		5			2	
2			2	3		
3	1			5	3	
4		3	5			4
5			1	6		
6		6			4	

9x9 & 12x12 [Not Always Square Number]?

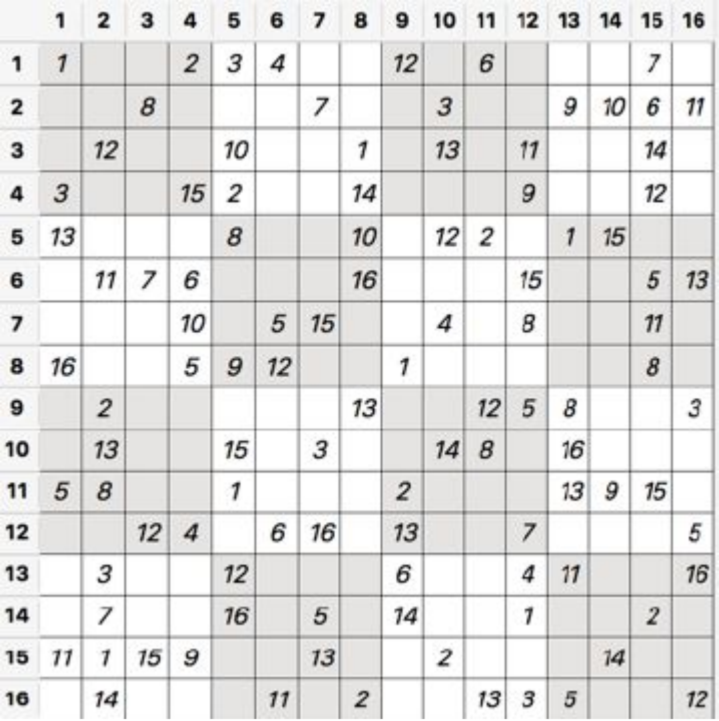


	1	2	3	4	5	6	7	8	9
1				1			3	2	
2				4	2		1		
3	4	1	2	7			6		
4	8								6
5				2	8	9			
6	5								9
7			7			2	9	3	8
8			4		3	1			
9		9	3			7			

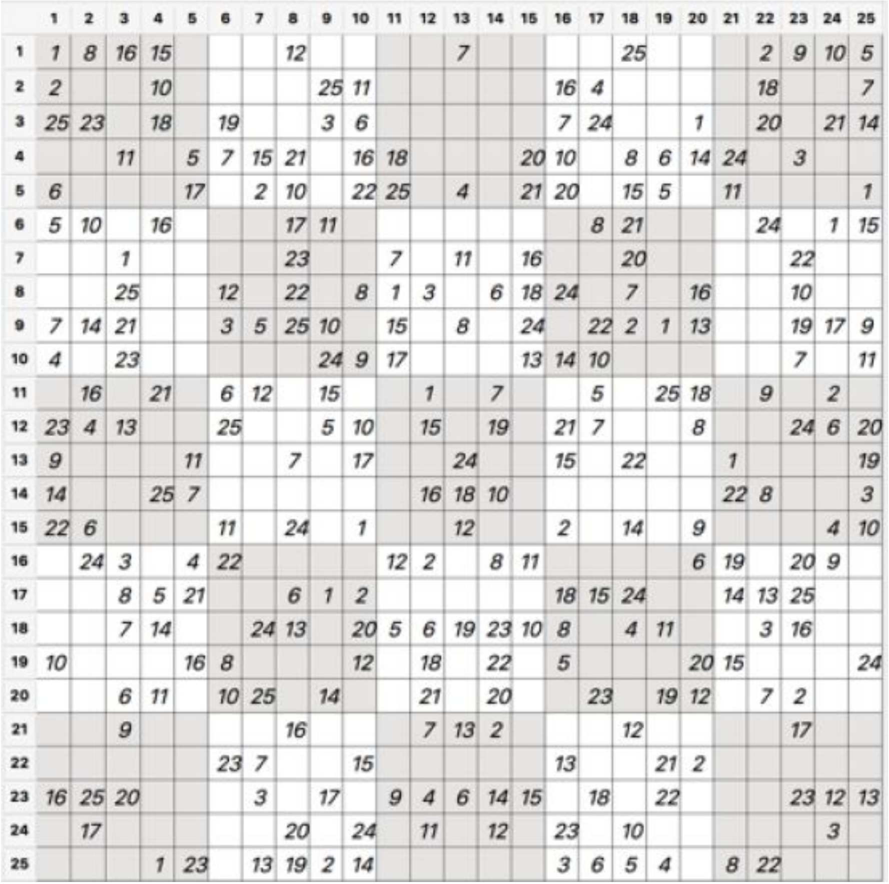


	1	2	3	4	5	6	7	8	9	10	11	12
1		1					4			7	3	
2	9			8		11		7				
3	6	7									11	
4	11		3		10				1		7	
5	5					3		4		10		
6				7				6		3		
7			8		1				11			
8			4		6							3
9				6				11	9	4		8
10		9					3				2	1
11					5		6		4			10
12	8	10	11			7					9	

16x16 & 25x25?



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1			2	3	4			12		6				7	
2			8				7			3			9	10	6	11
3		12			10			1		13		11			14	
4	3			15	2			14				9			12	
5	13				8			10		12	2		1	15		
6		11	7	6				16				15			5	13
7				10		5	15			4		8			11	
8	16			5	9	12			1						8	
9		2						13			12	5	8			3
10		13			15		3			14	8		16			
11	5	8			1				2				13	9	15	
12			12	4		6	16		13			7				5
13		3			12				6			4	11			16
14		7			16		5		14			1			2	
15	11	1	15	9			13			2				14		
16		14				11		2			13	3	5			12

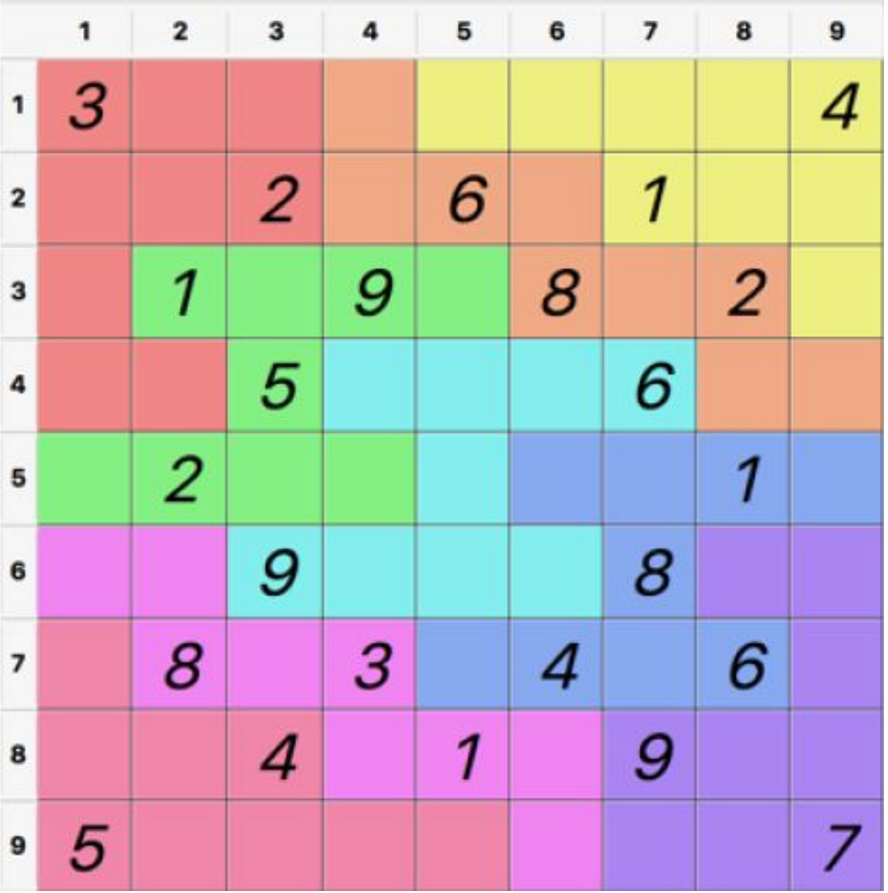


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	1	8	16	15				12					7					25				2	9	10	5
2	2			10					25	11						16	4					18			7
3	25	23		18		19			3	6						7	24			1		20		21	14
4			11		5	7	15	21		16	18				20	10		8	6	14	24		3		
5	6				17		2	10		22	25		4		21	20		15	5		11				1
6	5	10		16				17	11								8	21				24		1	15
7			1					23			7		11		16			20					22		
8			25			12		22		8	1	3		6	18	24		7		16			10		
9	7	14	21			3	5	25	10		15		8		24		22	2	1	13			19	17	9
10	4		23						24	9	17				13	14	10						7		11
11		16		21		6	12		15			1		7			5		25	18		9		2	
12	23	4	13			25			5	10		15		19		21	7			8			24	6	20
13	9				11			7		17			24			15		22			1				19
14	14			25	7							16	18	10							22	8			3
15	22	6				11		24		1			12			2		14		9				4	10
16		24	3		4	22					12	2		8	11					6	19		20	9	
17			8	5	21			6	1	2						18	15	24			14	13	25		
18			7	14			24	13		20	5	6	19	23	10	8		4	11			3	16		
19	10				16	8				12		18		22		5				20	15				24
20			6	11		10	25		14			21		20			23		19	12		7	2		
21			9					16				7	13	2				12					17		
22						23	7			15						13			21	2					
23	16	25	20				3		17		9	4	6	14	15		18		22				23	12	13
24		17						20		24		11		12		23		10						3	
25				1	23		13	19	2	14						3	6	5	4		8	22			

JIGSAW SUDOKU?

The only difference to the standard Sudoku is the shape of the inner blocks

- Grid size: 9 × 9
- Number of inner boxes: 9
- Inner box shape: Nonomino (polyomino of order 9)
- Number range: 1 to 9



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

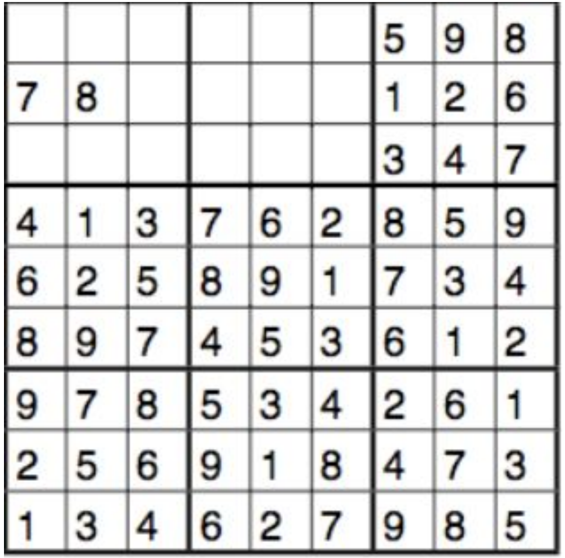
ARE SUDOKU ALWAYS SOLVABLE?

Answer is : **YES**

Unless it doesn't Violates the 3 Standard Rules of Sudoku [Row,Column,Grid]

Any Sudoku can have:

- ★ Unique Solution
- ★ Multiple Solution [Where we need to Guess]

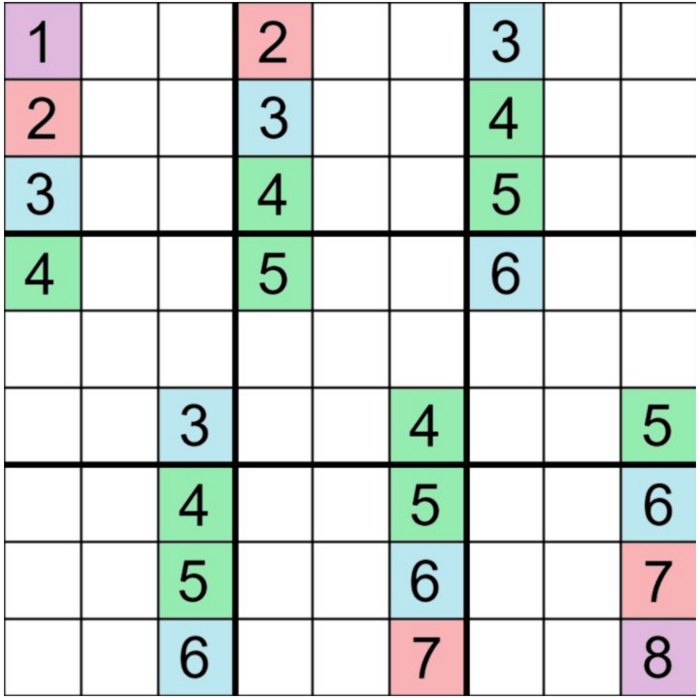


						5	9	8
7	8					1	2	6
						3	4	7
4	1	3	7	6	2	8	5	9
6	2	5	8	9	1	7	3	4
8	9	7	4	5	3	6	1	2
9	7	8	5	3	4	2	6	1
2	5	6	9	1	8	4	7	3
1	3	4	6	2	7	9	8	5

HOW MANY HINTS NEEDED TO SOLVE SUDOKU?

There are 26 types of **symmetry**, but they can only be found in about **0.005%** of all filled grids.

A puzzle with a unique solution must have at least **17 clues**, and there is a solvable puzzle with at most 21 clues for every solved grid.



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

WHAT ARE METHODS OF SOLVING SUDOKU?

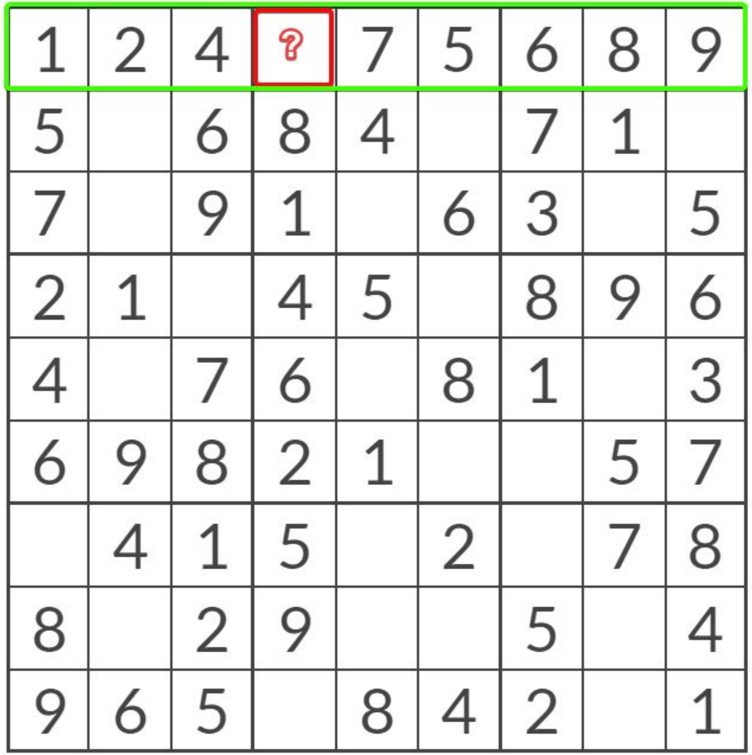
Sudoku puzzles are **NP-complete** problems and as such, their solution can be found by performing an **exhaustive search**.

1. **Pen and Paper Algorithm**
2. **Brute Force Search/BackTracking/Depth For Search**
3. **Crook's Algorithm**

PEN AND PAPER ALGORITHM

(1) Unique missing candidate

- Check if a Row/Column/Grid has only 1 Number Left

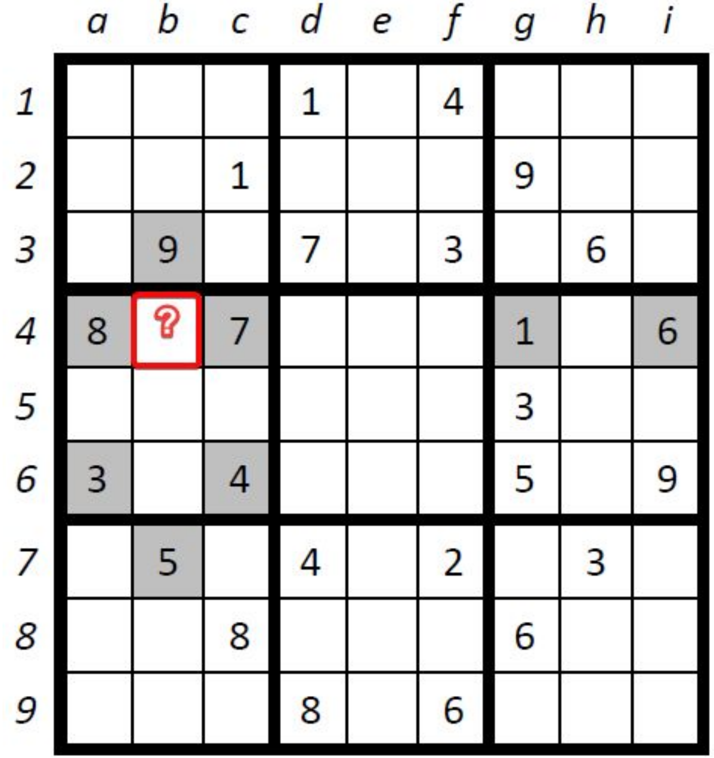


1	2	4	?	7	5	6	8	9
5		6	8	4		7	1	
7		9	1		6	3		5
2	1		4	5		8	9	6
4		7	6		8	1		3
6	9	8	2	1			5	7
	4	1	5		2		7	8
8		2	9			5		4
9	6	5		8	4	2		1

PEN AND PAPER ALGORITHM

(2) Naked Single

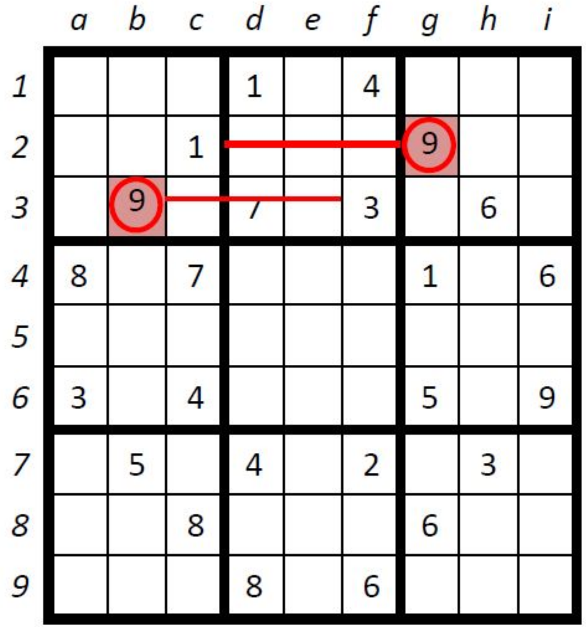
- Find a square that takes a single value, considering squares in the same row, column and box.



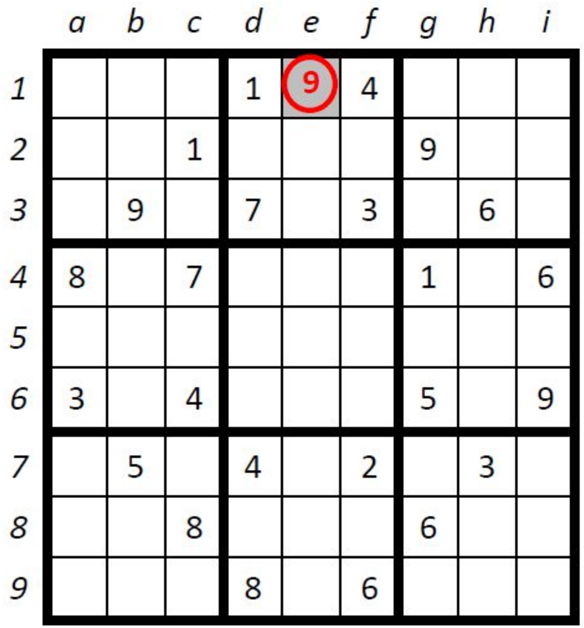
	a	b	c	d	e	f	g	h	i
1				1		4			
2			1				9		
3		9		7		3		6	
4	8	?	7				1		6
5							3		
6	3		4				5		9
7		5		4		2		3	
8			8				6		
9				8		6			

PEN AND PAPER ALGORITHM

(3) Hidden Single



	a	b	c	d	e	f	g	h	i
1				1		4			
2			1				9		
3		9		7		3		6	
4	8		7				1		6
5									
6	3		4				5		9
7		5		4		2		3	
8			8				6		
9				8		6			



	a	b	c	d	e	f	g	h	i
1				1	9	4			
2			1				9		
3		9		7		3		6	
4	8		7				1		6
5									
6	3		4				5		9
7		5		4		2		3	
8			8				6		
9				8		6			

FAKE OR NOT FAKE - QUESTION

“The difficulty of a sudoku puzzle is related to the number of clues it has”

Greater the Number of Clues, Lesser the Difficulty !

FAKE OR NOT FAKE?

FAKE OR NOT FAKE ANSWER

Answer is **Fake**! What?

The level of difficulty is defined by the **number of steps required to solve it**, as well as the **complexity of the techniques** needed for a person/an algorithm to solve the puzzle.

A bottleneck is a situation in which there is only a single move which will make it possible to continue solving. Thus, **the more bottlenecks in a puzzle, the harder it becomes**, even if the required solving techniques are easy, as many squares have to be visited in order to find that specific move.

FAKE OR NOT FAKE ANSWER

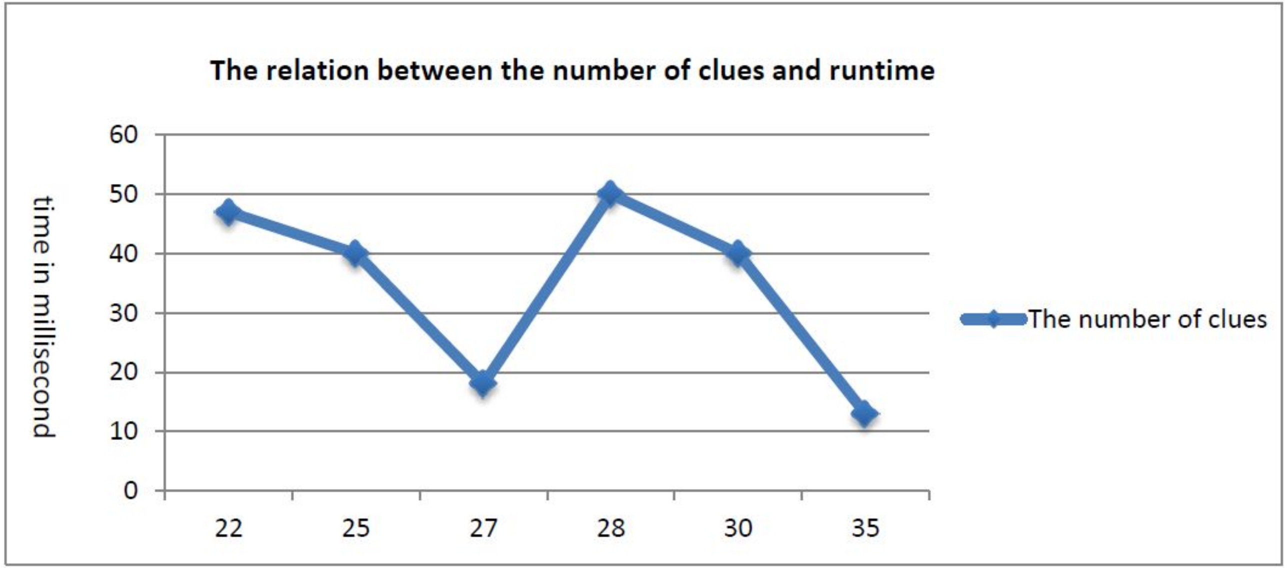


***Diagram 3**. This diagram shows the relationship between the number of clues (in the horizontal axis) and the run-time of the puzzles (in the vertical axis).*

FEATURES OF OUR PROJECT

Sudoku Generation

- Easy Level
- Medium Level
- Hard Level
- Random Level
- Sudoku with required number of missing squares

FEATURES OF OUR PROJECT

Sudoku Solving

- Solving by **Backtracking** Algorithm
- Solving by **Crook's** Algorithm
- **Time taken** for solving

BACKTRACKING

- Backtracking checks **every possible outcome** that can occur on the board and is terminated when correct output occurs in one of these outcomes.

- Backtracking is Also known as “**Complete Brute Force Search**” OR “**Depth For Search Traversal** Approach in Grid”

BACKTRACKING ALGORITHM

In simple terms, we first **start** with the **first empty space** on the sudoku board. We add the **first element** that can come in that space, let's say **1**.

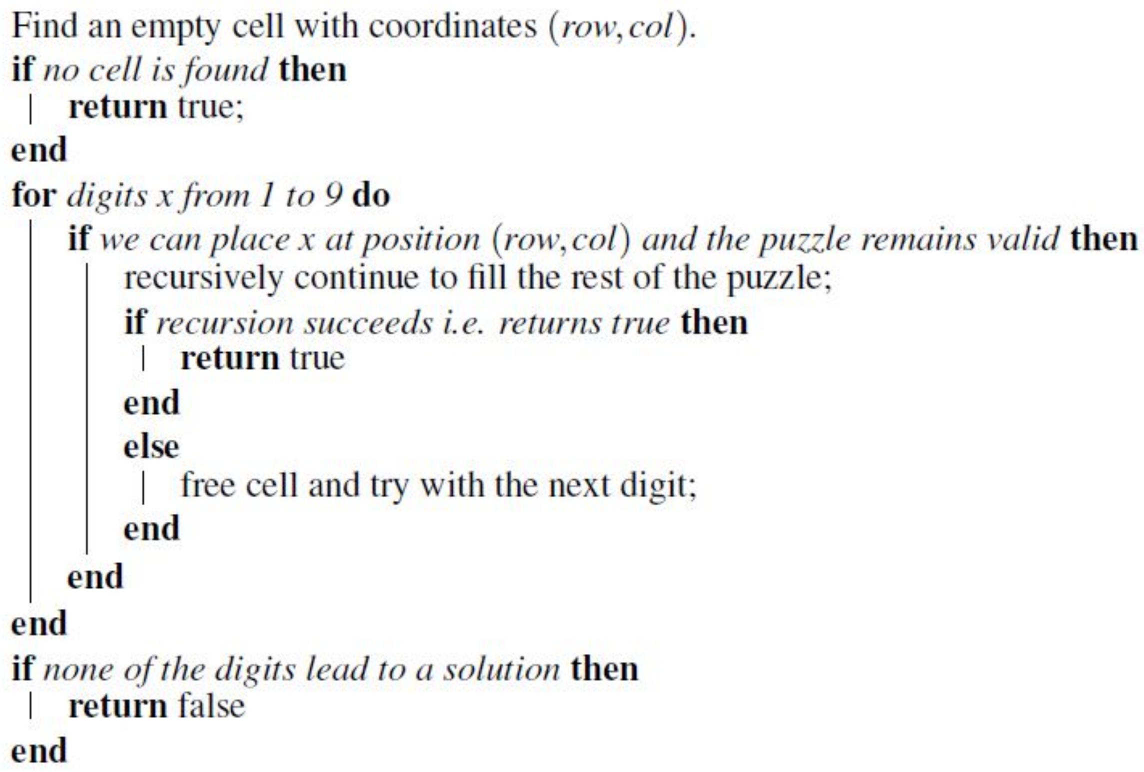
Next, we move on to the **second empty space,** and add the first element, **1**, to it.

If this is valid, it will go to **next space,** or else the element will become **2**. This **repeats for every element** that can be placed in that space.

This process is repeated for **every empty space** and if the sudoku is not solved, it goes to the *first empty space and places next possible element*, that is, 2, in our case.

This whole process is repeated until the sudoku is solved.

BACKTRACKING ALGORITHM



```
Find an empty cell with coordinates $(row, col)$.
if no cell is found then
    return true;
end
for digits x from 1 to 9 do
    if we can place x at position $(row, col)$ and the puzzle remains valid then
        recursively continue to fill the rest of the puzzle;
        if recursion succeeds i.e. returns true then
            return true
        end
        else
            free cell and try with the next digit;
        end
    end
end
if none of the digits lead to a solution then
    return false
end
```

BACKTRACKING - ANALYSIS

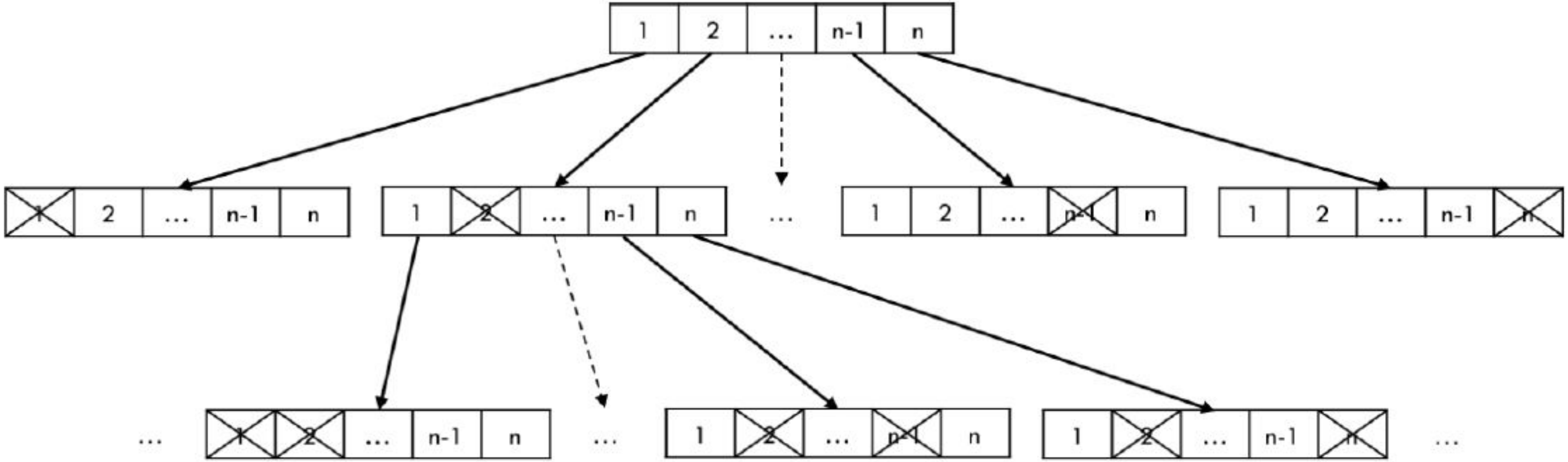
The time complexity for solving the sudoku is **$O(n^m)$**, where $m = n \times n$, where n is the number of squares in one side of the sudoku square. In classical sudoku, n = 9. Thus, time complexity for classical sudoku is **$O(9^{81})$**.

The recurrence relation for backtracking algorithm for classical sudoku can be written as

$$T(m) = 9*T(m-1) + O(1)$$

This is because, we have to check 9 values in a particular empty space.

BACKTRACKING - ANALYSIS



BACKTRACKING - ANALYSIS

The recurrence relation can be solved using **recursion tree** method.

We can say that at each level of the tree time taken would be **$n^{k-1}c$** , where **k** is the *current height of the tree* and **n** is the *number of square on one side of the sudoku board*. This would mean the time complexity would be

$$1 + n^1 + n^2 + \ldots + n^{k-1} = 1*(n^{k-1} -1)/n-1 = (1/(n-1))*n^k/n - (1/(n-1)) = (1/n*(n-1))*n^k - 1/(n-1) = \mathbf{O(n^k)}$$

Thus, the time complexity of solving a sudoku through backtracking would be **$O(n^k)$**.

CROOKS ALGORITHM

Crooks algorithm, which is an *imitation of pencil and paper algorithm*, is a simple algorithm which we use to solve the sudoku on our own.

This algorithm sounds simple, but it has its **limitations**. Sometimes, we get **incomplete answer to the sudoku** because of multiple values that can occur on a **particular empty space**.

CROOKS ALGORITHM

(1) Find all **forced numbers** in the puzzle

(2) Mark up the puzzle

(3) Search **iteratively for preemptive sets** in all rows, columns, and boxes—taking appropriate *crossout action* as each new preemptive set is discovered—until

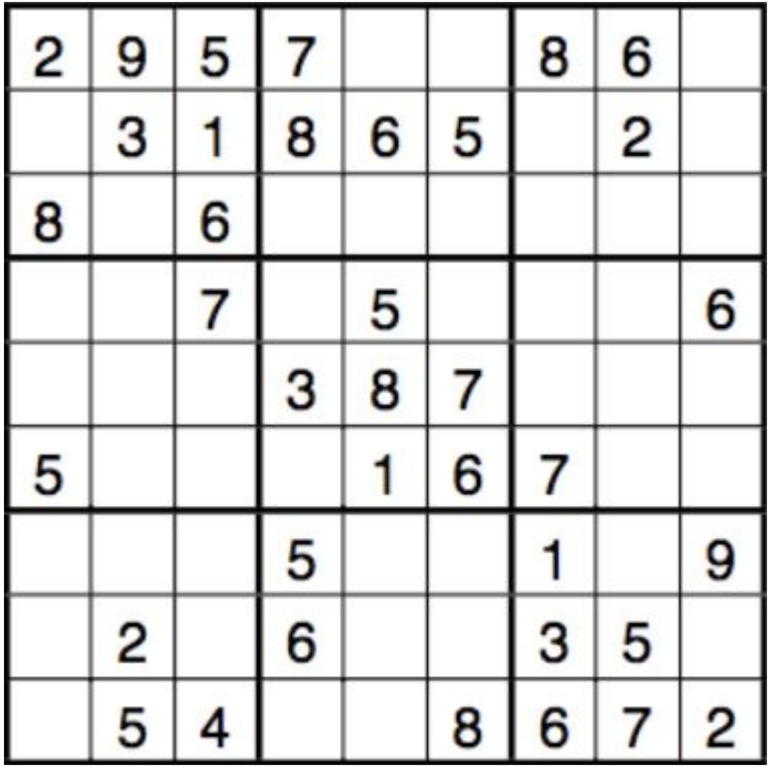
(4) either

  (a) **a solution is found**; or

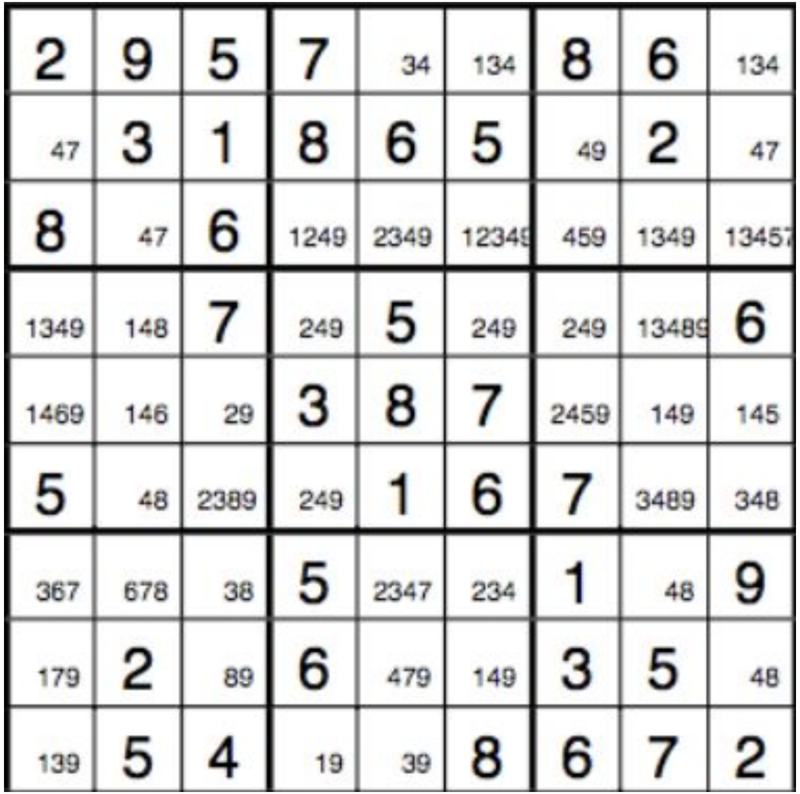
  (b) a **random choice** must be made for continuation.

(5) If 4(a), then **end**; if 4(b), then go to step 3.

EXAMPLE



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



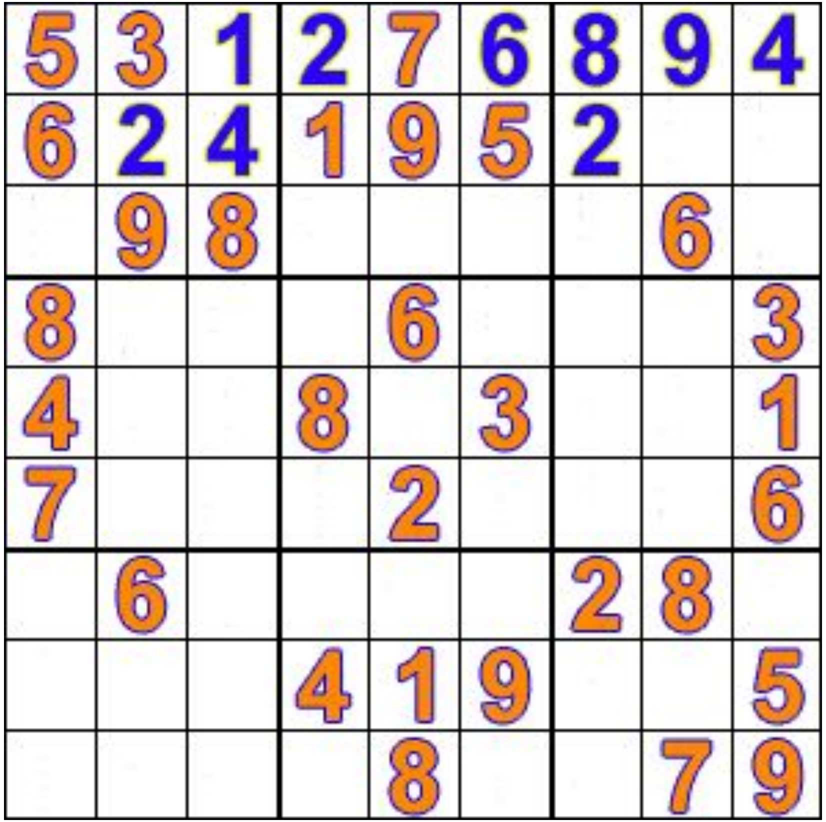
2	9	5	7	34	134	8	6	134
47	3	1	8	6	5	49	2	47
8	47	6	1249	2349	12349	459	1349	13457
1349	148	7	249	5	249	249	13489	6
1469	146	29	3	8	7	2459	149	145
5	48	2389	249	1	6	7	3489	348
367	678	38	5	2347	234	1	48	9
179	2	89	6	479	149	3	5	48
139	5	4	19	39	8	6	7	2

PSEUDO CODE [CROOKS ALGORITHM]

```
1 -> MARKUP

2 - > FIND SINGLETON

3 -> FIND PRE-EMPTIVE SET

4 -> ELIMINATE POSSIBLE NUMBERS OUTSIDE

PRE-EMPTIVE SET
```

LIVE DEMO

https://github.com/BhagyaRana/SUDOKU-SOLVER



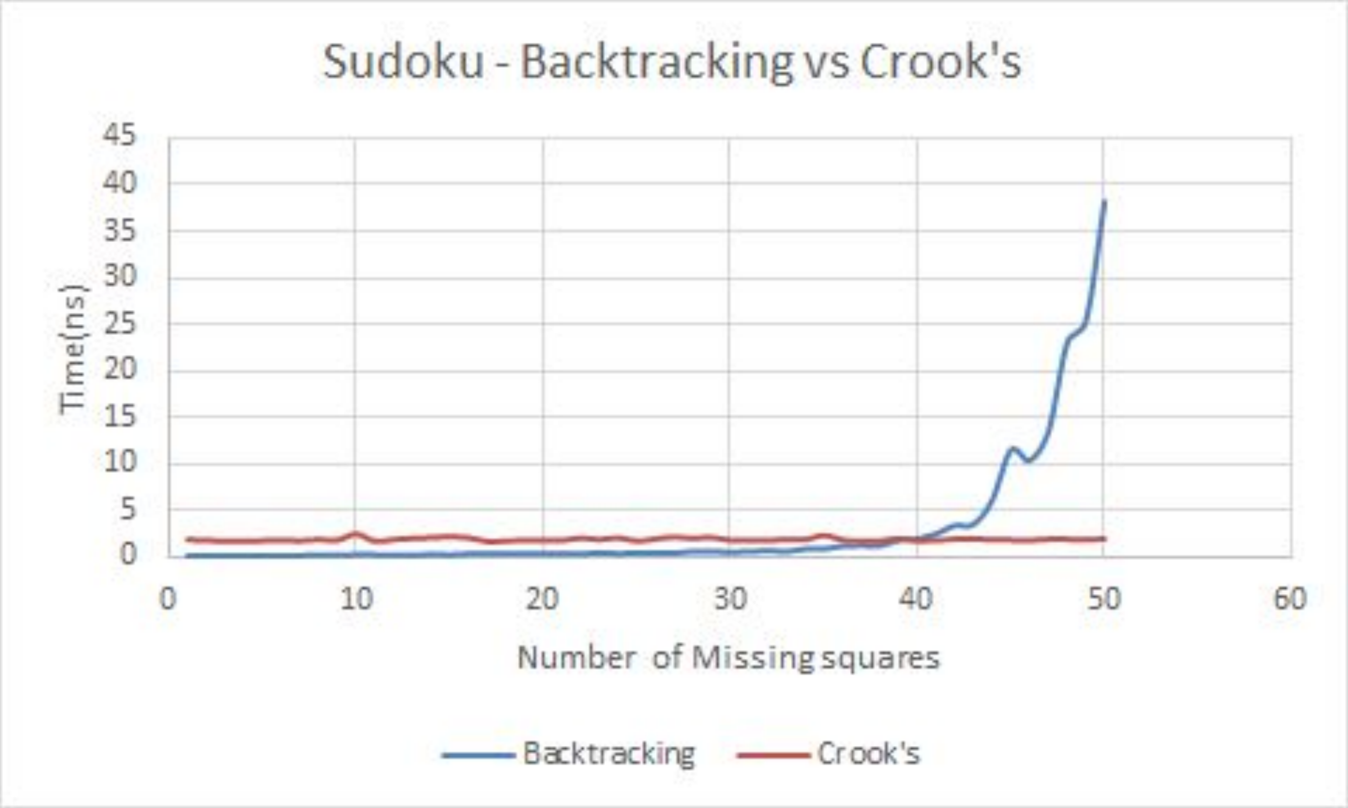
5	3	1	2	7	6	8	9	4
6	2	4	1	9	5	2		
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

COMPARISON

According to the data, **crooks algorithm** is much **faster** than backtracking algorithm as it eliminates the impossible outcomes and then solves the algorithm, as compared to backtracking which generates all possible sudoku matrices which takes more time.

On the other hand, complete version of Crooks algorithm is that, solve the puzzle, and if it is incomplete, complete it with backtracking.

COMPARISON


Sudoku - Backtracking vs Crook's
Time(ns)
45
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
60
Number of Missing squares
Backtracking
Crook's

TEST CASES FOR COMPARISON

Taking into consideration the fact that **number of clues and difficulty level** of a sudoku cannot be compared,

For **comparative performance analysis** of the two algorithms, average of the time taken by sudokus with same number of clues is taken.

WHICH IS BEST ALGORITHM?

- Crook’s on its own is an **incomplete** algorithm, but it imitates solving by pencil and paper like us. It Might Get Stuck Somewhere due to Own Wrong Assumption
- Backtracking checks all possible combinations but it provides **correct solution.** Although, it Takes More Time, But is Able to Provide Correct Answer
- Best Algorithm => **Backtracking Approach/ DFS Approach**

CONCLUSION

We have Successfully Learned the **Mathematical Logic** Working Behind the Sudoku and How to **Computationally Solve** this Problem in Step-By-Step Manner.

We have also Applied the **Development Knowledge** to Enhance the **User Experience** [Instead of Running on Terminal]

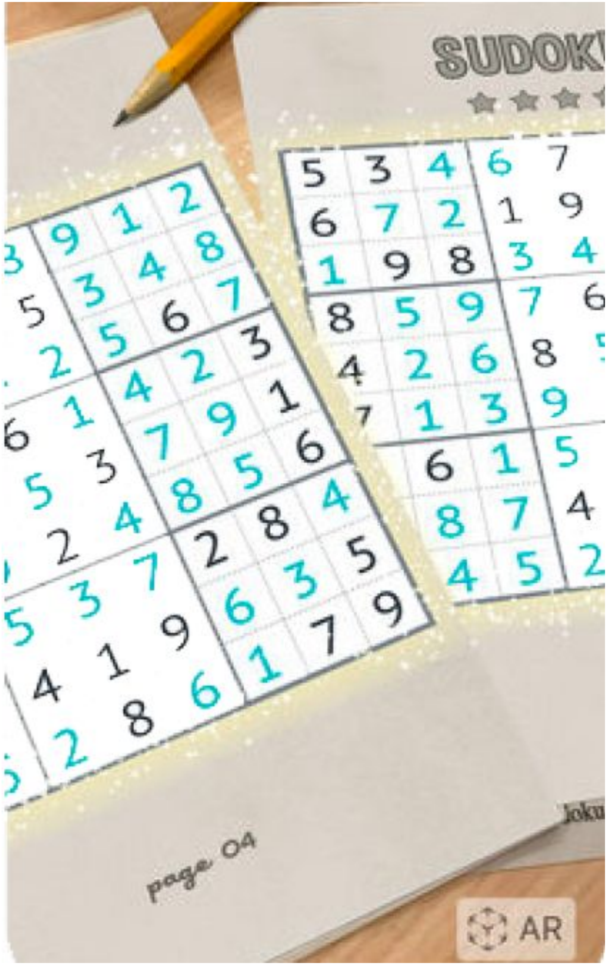
We have also Successfully **Understood** the **BackTracking Concept** and **Applied** it in Our **Project Sudoku Solver**!

FUTURE SCOPE

(1) This Project can be Integrated with **Computer Vision** [AR,OpenCV & Machine Learning] to **Extract Sudoku** from Newspaper and Solve it in Real Time & Display on Paper.

(2) Instant Help by **Giving Hints** and Users Progress while Playing

(3) Score System Based on **Time and Accuracy** [LeaderBoard and Game Development]



SOURCE CODE & GITHUB REPOSITORY

Source Code Link:

https://github.com/BhagyaRana/SUDOKU-SOLVER

Individual Contribution in Github Repository :

https://github.com/BhagyaRana/SUDOKU-SOLVER/graphs/contributors

CONTRIBUTIONS

Sr. No.	Admission No.	Name	Contribution
1	U19CS011	RAJ JIKADRA	Algorithm Implementation of Sudoku Solver, Comparative Performance Analysis of Algorithm, Report Analysis
2	U19CS012	BHAGYA RANA	Front-End & U.I. Build, Web App Infrastructure Developer, Project P.P.T .and Report Documentation, Video Presenter

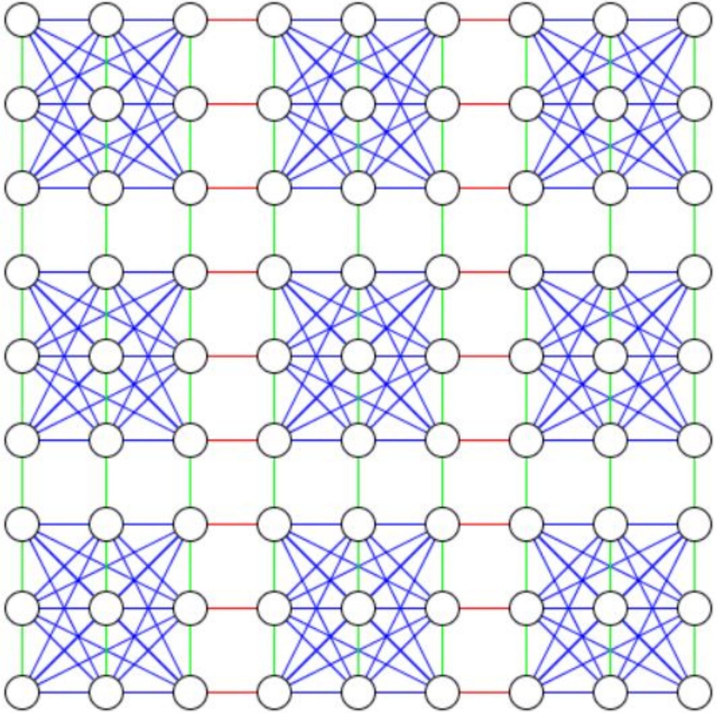
CONTRIBUTIONS

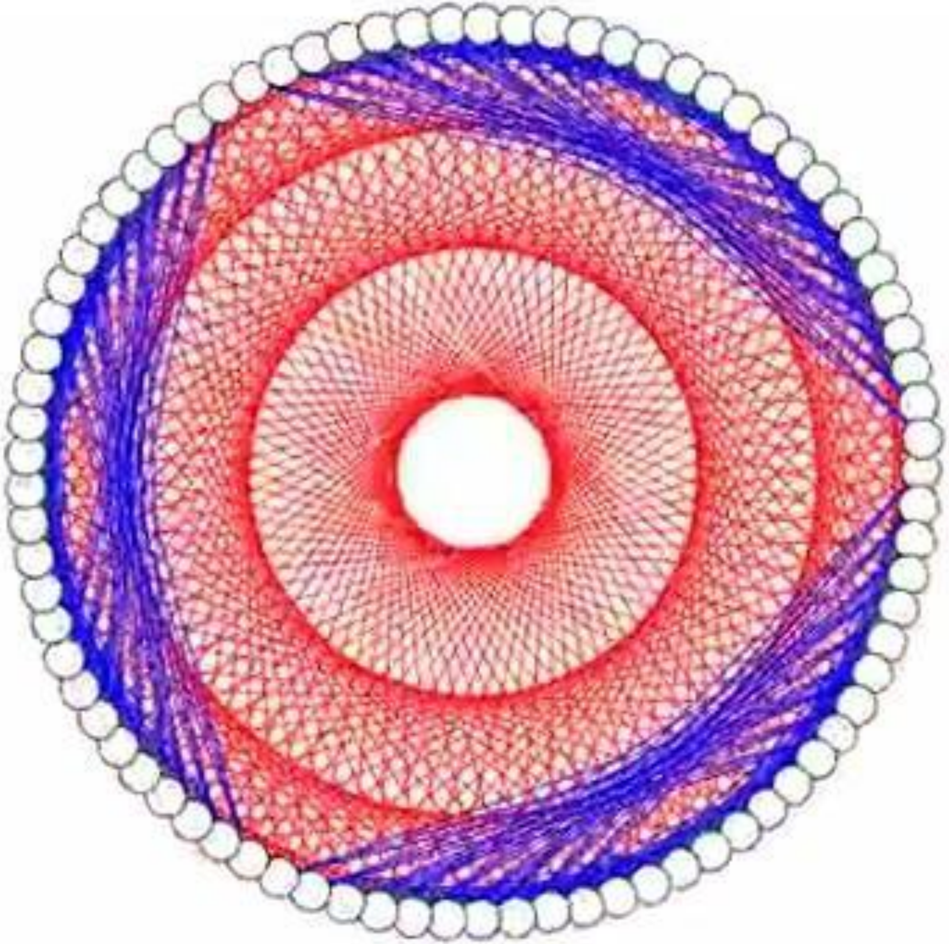
Sr. No.	Admission No.	Name	Contribution
3	U19CS049	DEV JARIWALA	Python Implementation of Algorithm, Presentation and Report Making, Video Editing
4	U19CS080	MUKTESHARYAN UPPALA	Backend (API Generation) for Web App, Python code for Generation of Sudoku, Algorithm Analysis, PPT and Report Proofreading, Video Presenter

CAN SUDOKU BE SOLVED USING GRAPH COLORING?

YES/NO?

Surprisingly, The Answer is **Yes**!





LET’S SOLVE IT!

Next Time, You see a Sudoku,

Think of it’s **Variation**, **Method of Solving** & **Analysis of Various Approach**!

The **simplicity of puzzle’s structure** and the low requirement of mathematical skills caused people to have **enormous interest** in accepting challenges to solve the puzzle

DO YOU ACCEPT THE CHALLENGE?