**SUDOKU USING BACKTRACKING**

# A PROJECT REPORT

***Submitted by***

**Utkarsh Gupta [RA2211003030362]**

**Sukriti Shukla [RA2211003030363]**

**Archi Jaiswal [RA2211003030369]**

*Under the guidance of*

# Mrs. Ambuja Kulshreshtha

(Assistant Professor, Department of Computer Science & Engineering)

***In partial fulfillment for the award of the degree of***

# BACHELOR OF TECHNOLOGY

In

**COMPUTER SCIENCE & ENGINEERING**

Of

**FACULTY OF ENGINEERING AND TECHNOLOGY**



Delhi NCR Campus, Modinagar, Ghaziabad (U.P.)

**APRIL 2024**

**SRM INSTITUTE OF SCIENCE**

**&TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

# BONAFIDE CERTIFICATE

Certify that this project report titled “**SUDOKU USING BACKTRACKING**” is the bonafide work of “ **Utkarsh Gupta [RA2211003030362] ,**

## Sukriti Shukla [RA2211003030363], Archi Jaiswal

**[RA2211003030369]**” who has completed the project work under my supervision. Certified further, that apparently the work detailed thus shapes no other task report or paper based on which a degree or grant was given on a prior event on this or some other candidate.

**SIGNATURE SIGNATURE**

(Mrs. Ambuja kulshreshtha) (Dr. Avneesh Vasistha)

**GUIDE HEAD OF**

**DEPARTMENT**

Assistant Professor

Dept. of Computer Science Dept. of Computer Science

& Engineering & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

# ABSTRACT

In the last decade, solving the Sudoku puzzle has become everyone’s passion. The simplicity of the puzzle s structure and the low requirement of mathematical skills caused people to have enormous interest in accepting challenges to solve the puzzle. Therefore, developers have tried to find algorithms to generate a variety of puzzles for human players so that they could be even solved by computer programming. In this essay, we have presented an algorithm called pencil-and-paper using human strategies. The purpose is to implement a more efficient algorithm and then compare it with another Sudoku solver named as brute force algorithm. This algorithm is a general algorithm that can be employed in any problem. The results have proved that the pencil-and-paper algorithm solves the puzzle faster and more effectively than the brute force algorithm.

# ACKNOWLEDGEMENT

We would like to thank our guide, Ms. Priyanka Gupta, for her valuable guidance, constant encouragement, personal caring, timely aid and offering timely assistance as well as providing us a wonderful environment in which to explore ourselves. Throughout the project, she has extended bright and agreeable assistance to us, despite her hectic schedule.

We would also want to express our gratitude to Dr. Avneesh Vasistha, Professor and Head, Department of B.Tech Second Year, SRM Institute of Science and Technology (Delhi-NCR Campus), for his helpful recommendations and support during the project's duration. We would like to express our gratitude to the personnel and students at the SRM Institute of Science and Technology's Computer Science and Engineering Department for their assistance during our investigation. Finally, we want to express our gratitude to our parents, family, and friends for their unwavering love, constant assistance, and unwavering support.

Utkarsh Gupta [RA2211003030362]

Sukriti Shukla [RA2211003030363]

Archi Jaiswal [RA2211003030369]

# TABLE OF CONTENT

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** | 3 |
|  | **ACKNOWLEDGEMENT** | 4 |
|  | **1) LIST OF FIGURES** | 7 |
|  | **2) LIST OF SYMBOLS & ABBREVIATIONS** | 7 |
| **1** | **INTRODUCTION** | 6 |
| **1.1** | **PROBLEM EXPLAINATION** | 6-7 |
| **2** | **LITERARY SURVEY** | 8 |
| **3** | **SYSTEM ANALYSIS** | 9 |
| **4** | **DESIGN TECHNIQUES** | 11 |
| **5** | **ALGORITHM FOR PROBLEM** | 12 |
| **6** | **SOURCE CODE** | 13-14 |
| **7** | **SAMPLE OUTPUT** | 15 |
| **8** | **CONCLUSION** | 16 |
| **9** | **REFERENCES** | 17 |
|  |  |  |

**CHAPTER-1**

**INTRODUCTION**

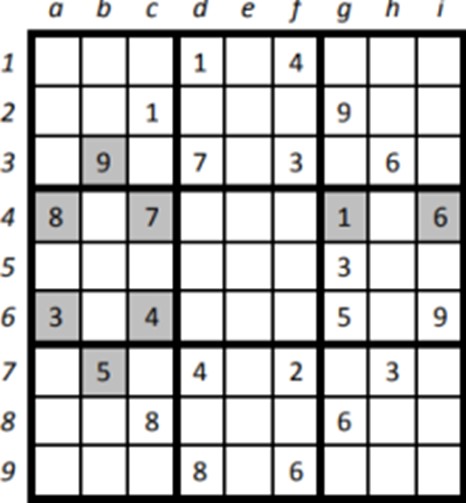
Solving Sudoku has been a challenging problem in the last decade. The purpose has been to develop more effective algorithm in order to reduce the computing time and utilize lower memory space. This essay develops an algorithm for solving Sudoku puzzle by using a method, called pencil-and-paper algorithm. This algorithm resembles human methods, i.e. it describes how a person tries to solve the puzzle by using certain techniques. Our ambition is to implement the pencil-and-paper algorithm by using these techniques. There are currently different variants of Sudoku such as 4 4 grids, 9 9 grids and 16 16 grids. This work is focused on classic and regular Sudoku of 9 9 board, and then a comparison is performed between the paper-and-pencil method and Brute force algorithm. Hopefully, by doing this work we might be able to answer the following questions: How does the pencil-and-paper algorithm differ from the Brute force algorithm Which one of them is more effective Is it possible to make these algorithms more efficient.

## 1.1 Problem Explaination

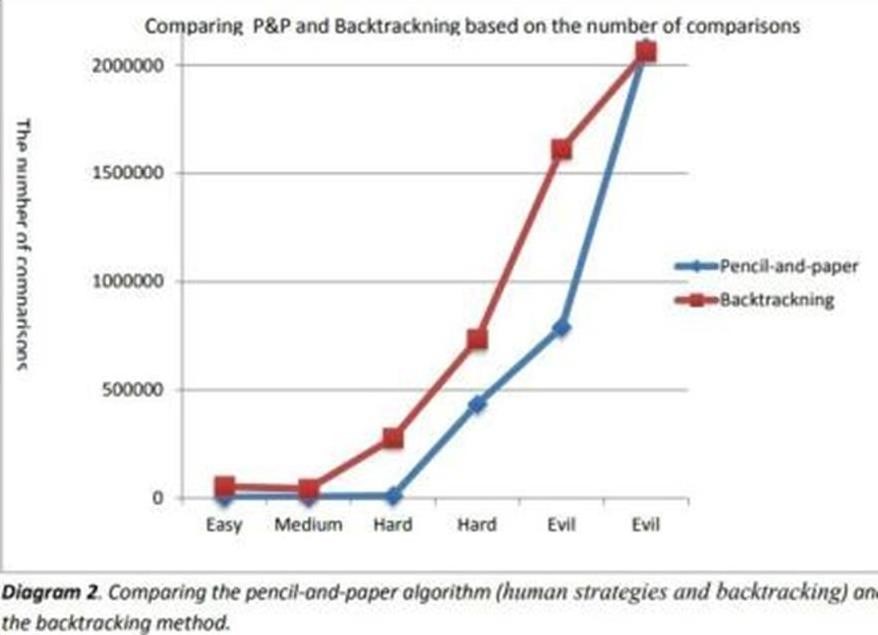
Problem Explaination Sudoku is a puzzle played on a partially filled 9 x 9 grid. The task is to complete the assignment using numbers from 1 to 9 such that the entries in each row, each column and each ma or 3 x 3 block are pairwise different. Below are given two tables the first is showing a Sudoku Problem and the second a solution to this problem. By choosing a good viewpoint, most instances of the Sudoku Puzzle can be solved without search.

The structure of the puzzle is very simple, especially the classic puzzle. This essay is mainly focused on classic puzzle of a 9 9 grid. There already exist a number of digits in the board that make the puzzle solvable. It means that some numbers are already placed in the Sudoku board before starting playing. The board consists of 1 cells, which is divided into nine 3 3 sub boards and each 3 3 sub board is called box or region . The main concept of the game is to place numbers from 1 to 9 on a 9 9 board so that every row, column and box contains any numbers but once. This means that no number is repeated more than once.

There are numerous ways to solve a Sudoku puzzle. Backtracking is the most commonly used and easy to understand, though probably not the most efficient. The constraints given at the beginning of the puzzle narrow down the available numerals that can be placed in the empty boxes. For each blank, the solver can build a candidate bank, a list of all the numbers that meet the known constraints. When the candidate bank has been narrowed to one for a particular space, that numeral can be added to the grid. Each (correct) answer adds further constraints for the remaining spaces in the same row, column, or box, helping to narrow the candidate banks for the remaining questions.



An example of 9x9 sudoku



## CHAPTER-2

### LITERARY SURVEY

Solving Sudoku using backtracking is a well-studied topic in computer science and artificial intelligence. Backtracking is a technique for finding solutions to problems by incrementally building candidates for the solution, and if a candidate turns out to be invalid, the algorithm backs up and tries a different one. Here's a literature survey covering various aspects of solving Sudoku with backtracking:

* **Peter Norvig's Sudoku Solver:** Peter Norvig, Director of Research at Google, presented a simple but effective Sudoku solver using constraint propagation and search techniques.
* **Donald Knuth's Dancing Links Algorithm:** Donald Knuth introduced the Dancing Links technique for exact cover problems, which includes Sudoku solving.
* **Optimization Techniques:** Various optimization techniques have been proposed to improve the efficiency of backtracking-based Sudoku solvers.
* Parallel and Distributed Solvers: Researchers have explored parallel and distributed approaches to Sudoku solving to leverage multiple processors or computers.
* **Hybrid Approaches:** Hybrid approaches combine backtracking with other techniques such as genetic algorithms, simulated annealing, or neural networks.
* **Human-Solving Strategies:** Studying human-solving strategies can provide insights into improving computer algorithms. Research has been conducted to analyze common strategies used by humans to solve Sudoku puzzles and to incorporate these strategies into automated solvers.

**CHAPTER-3**

# SYSTEM ANALYSIS

A system analysis of solving Sudoku using backtracking involves breaking down the problem-solving process into its constituent parts and analyzing each component's functionality, efficiency, and interactions within the system. Here's how such an analysis might proceed:

**1. Problem Definition:**

* Define the problem: solving a Sudoku puzzle involves filling in a 9x9 grid with digits from 1 to 9 so that each row, each column, and each of the nine 3x3 sub grids contains all of the digits from 1 to 9 without repetition.
* Define the objective: find a valid solution for the given Sudoku puzzle. - Define constraints: the solution must adhere to the rules of Sudoku, ensuring that no digit is repeated within rows, columns, or sub grids.

**2. Algorithm Selection:**

- Choose an appropriate algorithm: Backtracking is a widely used algorithm for solving constraint satisfaction problems like Sudoku. It systematically explores the solution space by making a series of choices and backtracking when a choice leads to an invalid solution.

**3. System Architecture:**

* Design the overall system architecture: The system architecture should include components for representing the Sudoku puzzle, implementing the backtracking algorithm, and handling user input/output.
* Define interfaces between components: Specify how different components interact with each other. For example, the backtracking algorithm might receive the initial puzzle state and return the solved puzzle.

**4. Data Structures:**

* Choose suitable data structures: Represent the Sudoku grid and its constraints efficiently. Common data structures include arrays, matrices, and sets.
* Optimize data structures for performance: Use data structures that enable fast access and manipulation of puzzle elements to minimize solution time.

**5. Algorithm Implementation:**

* Implement the backtracking algorithm: Translate the backtracking algorithm into code, taking into account optimizations such as constraint propagation, heuristic selection of next moves, and pruning of search space.
* Test the algorithm thoroughly: Develop test cases to verify that the algorithm produces correct solutions for various input puzzles and benchmark its performance.

**6. Performance Analysis:**

* Analyse time complexity: Evaluate the worst-case time complexity of the backtracking algorithm in terms of the number of recursive calls and the size of the solution space.
* Measure solution time: Experimentally measure the time taken by the algorithm to solve Sudoku puzzles of different difficulty levels.
* Analyse space complexity: Evaluate the memory requirements of the algorithm, including the space needed for data structures and the call stack during recursion.

## CHAPTER -4

### DESIGN TECHNIQUES

The backtracking method, which is similar to the human strategy (guessing), is used as a help method to the pencil-and-paper algorithm. In other words, if the puzzle cannot be filled when using the unique missing method and the naked single method, the backtracking method will take the puzzle and fill the rest of empty squares. The backtracking method find empty square and assign the lowest valid number in the square once the content of other squares in the same row, column and box are considered. However, if none of the numbers from 1 to 9 are valid in a certain square, the algorithm backtracks to the previous square, which was filled recently.

The above-mentioned methods are an appropriate combination to solve any Sudoku puzzles. The naked single method can find quickly single candidates to the empty squares that needed only one single value. Since the puzzle comes to its end solution the unique missing method can be used to fill rest of the puzzles. Finally, if either method fills the board the algorithm calls the backtracking method to fill the rest of the board.

## CHAPTER-5

### ALGORITHM FOR THE PROBLEM

The unique missing method and the naked single method are able to solve all puzzles with easy and medium level of difficulties. In order to solve puzzles with even more difficult levels such as hard and evil the backtracking method has been used to complete the algorithm. A human player solves the puzzle by using simple techniques. If the puzzle is not solvable by using the techniques the player then tries to fill the rest of the empty squares by guessing.

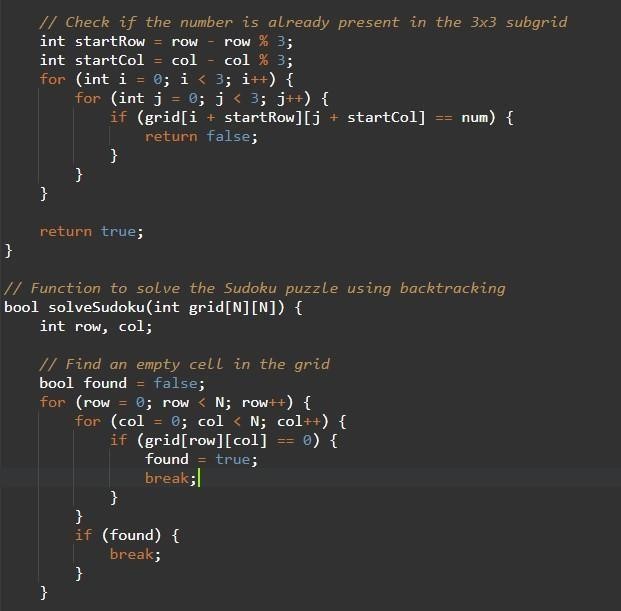
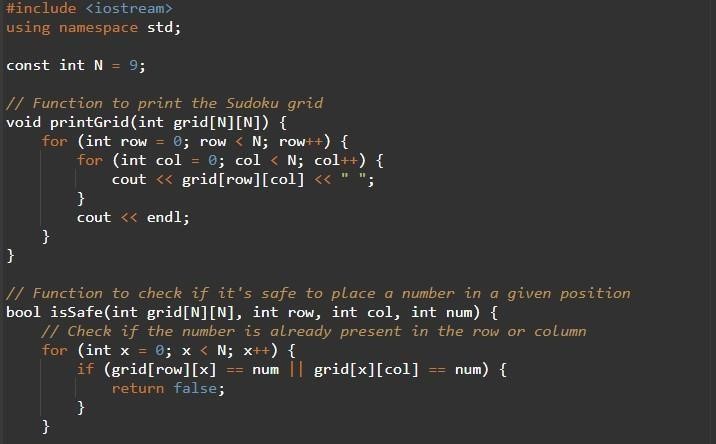
The backtracking method, which is similar to the human strategy (guessing), is used as a help method to the pencil-and-paper algorithm. In other words, if the puzzle cannot be filled when using the unique missing method and the naked single method, the backtracking method will take the puzzle and fill the rest of empty squares.

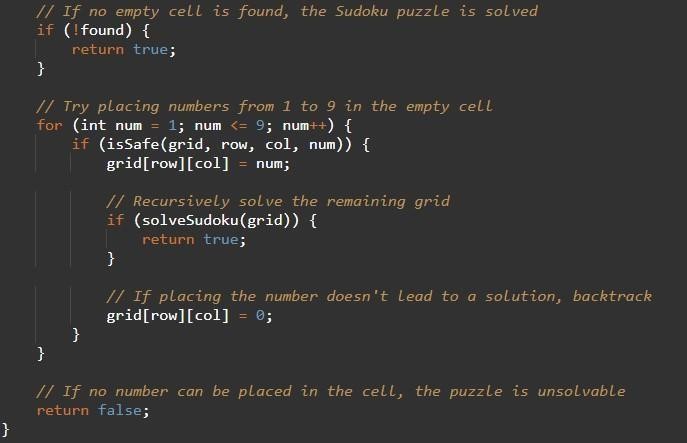
**Start from any cell and do the following steps recursively:**

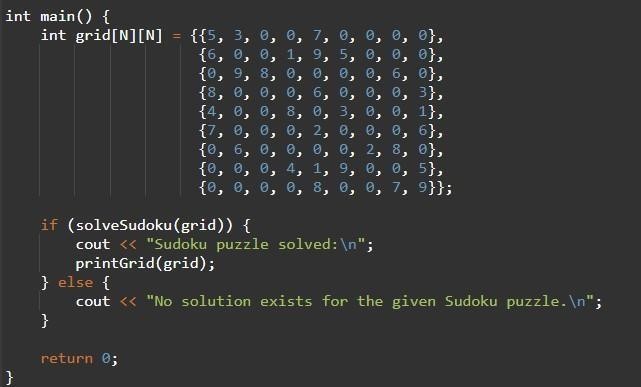
**1.** If the cell is empty, add a number that is not constrained.

1. If it is impossible to add a number due to constraints, report failure.
2. Else start a new thread on a new cell, starting from step 2. i. If this thread reports a failure, repeat step 2 with a new number (and exhaust the old number) ii. If this thread reports success, report success, since we can assume that the last cell has been successfully filled.
   1. If the cell is filled, then skip this cell and start a new thread on a new cell, starting from step2.
   2. If the algorithm has managed to move beyond the bounds of the board, report success.

**CHAPTER –6 SOURCE CODE**

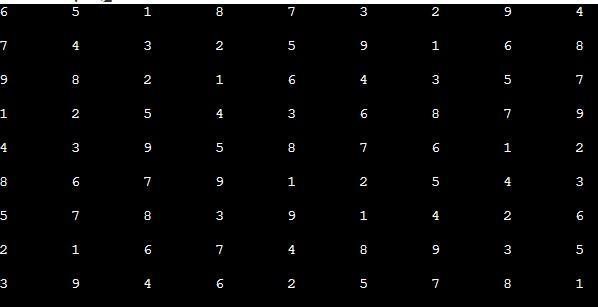






**CHAPTER -7**

**SAMPLE OUTPUT**



## CHAPTER -8

### CONCLUSION

This study has shown that the pencil-and-paper algorithm is a feasible method to solve any Sudoku puzzles. The algorithm is also an appropriate method to find a solution faster and more efficient compared to the brute force algorithm. The proposed algorithm is able to solve such puzzles with any level of difficulties in a short period of time (less than one second). The testing results have revealed that the performance of the pencil-and-paper algorithm is better than the brute force algorithm with respect to the computing time to solve any puzzle. The brute force algorithm seems to be a useful method to solve any Sudoku puzzles and it can guarantee to find at least one solution. However, this algorithm is not efficient because the level of difficulties is irrelevant to the algorithm. In other words, the algorithm does not adopt intelligent strategies to solve the puzzles. This algorithm checks all possible solutions to the puzzle until a valid solution is found which is a time consuming procedure resulting an inefficient solver. As it has already stated the main advantage of using the algorithm is the ability to solve any puzzles and a solution is certainly guaranteed. Further research needs to be carried out in order to optimize the pencil-and-paper algorithm. A possible way could be implementing of other human strategies (x-wings, swordfish, etc.).Other alternatives might be to establish whether it is feasible to implement an algorithm based only on human strategies so that no other algorithm is involved in the pencil-and-paper algorithm and also make sure that these strategies can solve any puzzles with any level of difficulties

**CHAPTER-9 REFERENCES**

1) Wikipedia [cited 2013 February 21],

Web site:<http://en.wikipedia.org/wiki/Sudoku>2) Home Of Logic Puzzles [cited 2013 February 22],

Web Page: http://www.conceptispuzzles.com/index.aspx?uri=puzzle/sudoku/ classic

1. J.F. Crook, A pencil and paper algorithm for solving Sudoku Puzzles, [Cited 2013 February 24], Winthrop University,

Webpage:<http://www.ams.org/notices/200904/tx090400460p.pdf>

1. A.S. Showdhury, S. Skher Solving Sudoku with Boolean Algebra [Cited 2013 February 24], International Journal of Computer Applications, Peer-reviewed Research,

Webpage: [http://research.ijcaonline.org/volume52/number21/pxc3879024.p d](http://research.ijcaonline.org/volume52/number21/pxc3879024.p)f

1. N. Pillay, Finding Solutions to Sudoku Puzzles Using Human Intuitive Heuristics, South African Research Articles,

Webpage: [http://sacj.cs.uct.ac.za/index.php/sacj/article/viewArticle/111 \](http://sacj.cs.uct.ac.za/index.php/sacj/article/viewArticle/111)

1. Ch. Xu, W. Xu, The model and Algorithm to Estimate the Difficulty Levels of Sudoku Puzzles, Journal of Mathematics Research,2009

Webpage: [http://journal.ccsenet.org/index.php/jmr/ a](http://journal.ccsenet.org/index.php/jmr/)rticle/ viewFile/3732/3336

1. T.Davis,TheMathematicsofSudoku(<http://www.geometer.org/in>dex.html),ResearchArticle,

Webpage:<http://share.dschola.it/castigliano/ips/Documentazione>

%20Progetto/Materiale% 20Didattico/Matematica/1E/sudoku.pdf