ARTIFICIAL INTELLIGENCE MINI PROJECT

Shivendra Pratap Singh (RA1911028010017)

Vaibhav Sharma (RA1911028010018)

Utkarsh Rastogi (RA1911028010019)

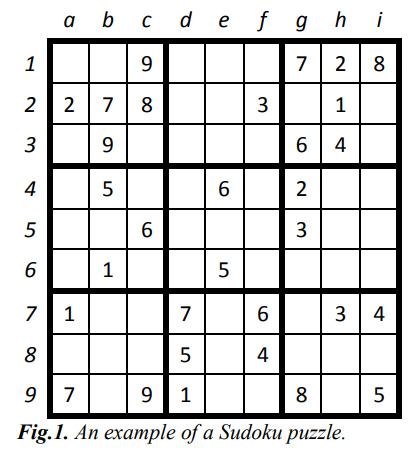
Problem Statement

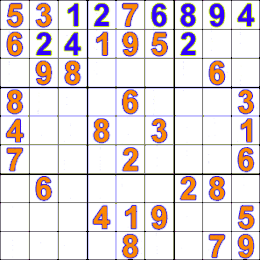
Currently, Sudoku puzzles are becoming increasingly popular among people all over the world. The game has become popular now in a large number of countries and many developers have tried to generate even more complicated and more interesting puzzles. Today, the game appears in almost every newspaper, in books and in many websites. In this essay we present a Sudoku Solver named as pencil-and-paper algorithm using simple

rules to solve the puzzles. The pencil-and-paper algorithm is formulated based on human techniques. This means that the algorithm is implemented based on human perceptions. Therefore the name of the solver is pencil-and-paper algorithm. The Brute force algorithm is

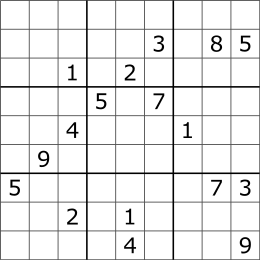
then used to compare with this algorithm in order to evaluate the efficiency of the proposed algorithm. The brute force is a general algorithm that can be applied to any possible problem. This algorithm generates any possible solutions until the right answer is found. The following subsections describe the problem statement, the purpose of this project, and the

abbreviations and the definitions.

Solving Sudoku has been a challenging problem in the last decade. The purpose has been to develop more effective algorithms in order to reduce the computing time and utilize lower memory space. This essay develops an algorithm for solving Sudoku puzzles by using a backtracking algorithm. This algorithm resembles human methods, i.e. it describes how a person tries to solve the puzzle by using certain techniques. 



A Sudoku (top) being solved by backtracking. Each cell is tested for a valid number, moving "back" when there is a violation, and moving forward again until the puzzle is solved.



A Sudoku designed to work against the brute force algorithm.

Objective

The objective of the program is to implement backtracking algorithm to solve sudoku solving program

Purpose of the problem statement

The social benefit of this project is that it can help you learn to solve Sudoku. The main benefits of Sudoku:

1. Improves concentration

It is impossible to solve a Sudoku puzzle without concentration. Since this game requires logical thinking, an interruption can break the chain of thought and force the player to restart their analysis. The frustration of having to constantly go back to square one in order to progress will eventually train the brain to block any source of distraction.

The more puzzles you play, the more absorbed in your task you will be each time, improving your concentration skills step by step.

This better-developed skill will not only be felt when playing Sudoku but will also transpire to other activities in your life, be it at work, studying or performing a task that requires your full attention.

2. Helps to reduce anxiety and stress

The two big bad words of today’s society can be tamed by a simple numbers puzzle. One of the benefits of Sudoku is that it requires the player to concentrate on the grid and use logical thinking to find the solution for each cell. While doing this, the brain becomes fully focus on the task at hand rather than the source of stress and anxiety.

This break can be just enough for the player to regain their sense of balance and become calmer. Once the puzzle is over, they might even find that the task or the problem that generated so much anxiety is not as daunting as initially seemed.

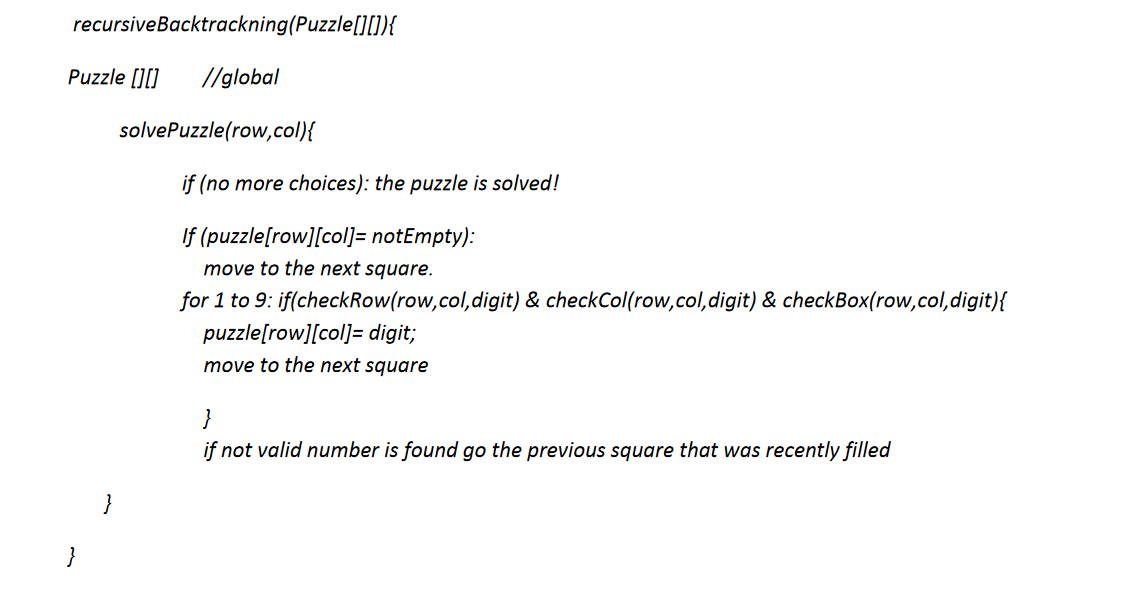
3. Promotes a healthy mindset

When the brain is not stimulated, it tends to dwell on negative thoughts and infuse the person with an overall sense of unhappiness.

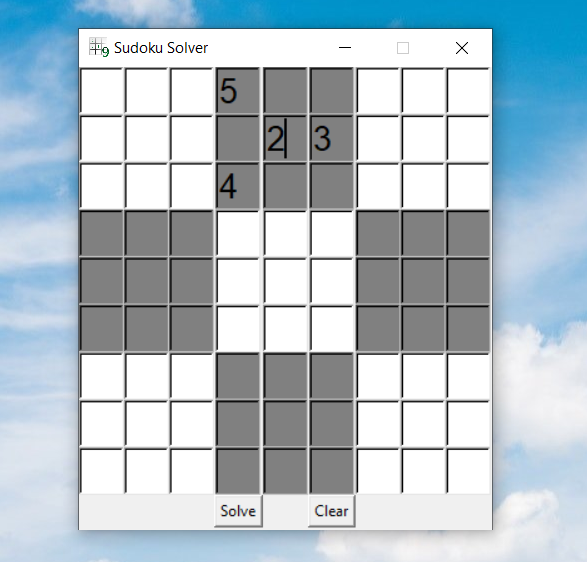
Just as exercising can boost your vitality, so does playing challenging thinking games like Sudoku. A fitter and happier brain is the first step to regard the world and your life with a healthier and more positive mindset.

Technical Depth

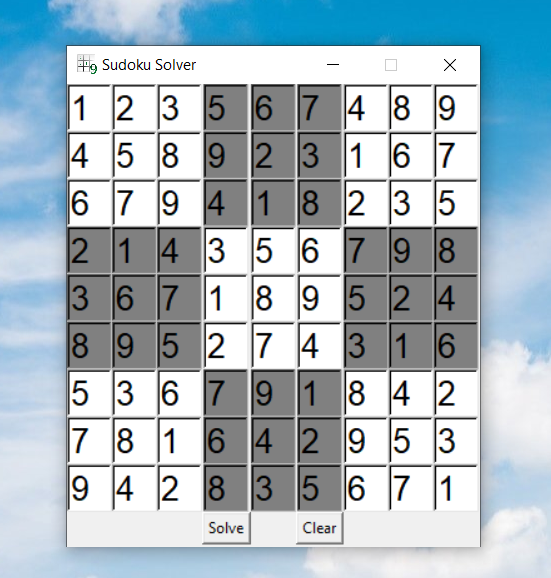
As mentioned before, we have used a Backtracking algorithm in the program. The algorithm is as follows.



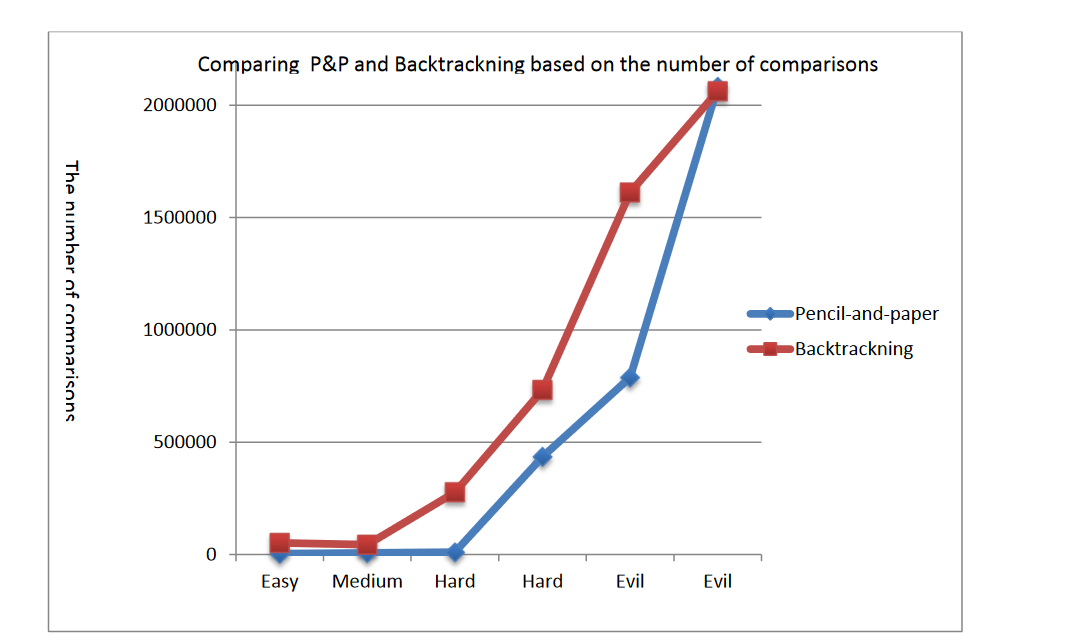
Execution



BEFORE AUTOSOLVING



AFTER AUTOSOLVING



Solving Sudoku is one of NP-complete problems

and it says that Sudoku algorithms do not scale well to larger boards and puzzles, for example

10000X10000 grids are not feasible. If the size of input to Sudoku solver goes to infinity the time of complexity will increase exponentially. However, when solving the puzzles with limited input size such as 9X9 grids, it is feasible because they can be solved in polynomial time.

GitHub Link: The project can be found on the following link - [Sudoku Solver](https://github.com/RastogiUtkarsh/SudokoSolver)