

# Sudoku Solver Using the Backtracking Algorithm

Kevin Cheng, Dominick Sy, Guntej Ghuman, Chris Nelson

**Abstract**— This document is meant to show our implementation of the backtracking algorithm used to solve Sudoku puzzles. Since this algorithm isn't particularly complicated once understood, we also implemented a file i/o system for ease of use for the end user and to help us learn more about the Python environment.

**Keywords**— Python, Algorithm, File I/O, Backtracking, Sudoku

## I. VI INTRODUCTION

The backtracking algorithm is one that is commonly used for solving puzzles and any problems that have constraints that must be satisfied before the answer is valid. Compared to the basic brute force approach, backtracking is significantly faster as it can dynamically revise its previous answers without having to start all over again, thus giving it its name.

## II. HOW THE PROGRAM WORKS

The program works via a simple file i/o system. It will read the Input.csv file and grab the numbers for the program. There the program applies a backtracking algorithm and outputs the answer into a formatted chart in the command line interface.

## III. PROGRAM INSTALLATION

Program installation is very easy. The user downloads the project folder and unzips all the contents. Once it's done unzipping the user can view the source code by navigating towards the src folder. The program code can be accessed and modified through any text editor (eg: Notepad, Python IDLE, Wordpad, etc).

If the user wants to run the application, there is a calculate.exe file outside the src folder. Clicking the exe file will load up the same sudoku csv file from the folder, calculate it for you and output back the answered board.

## IV. ENVIRONMENT SETUP

The test environment setup for this program is as follows:

*Hardware:*

- Operating System (Windows, other operating systems will need the program to be compiled again.)
- Modern Processor (i3 or above)
- 4 gb RAM or above

*Software:*

- Python IDE: PyCharm
- Test .csv files

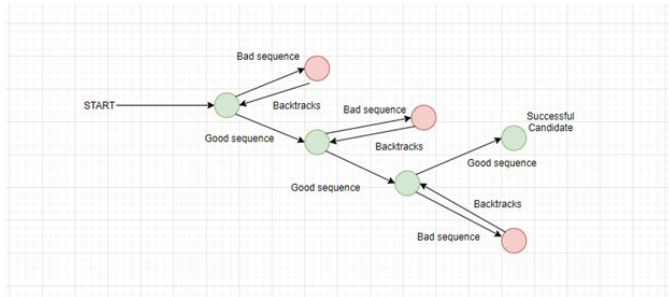
## V. EXPLANATION OF BACKTRACKING

Backtracking works by building possible solutions starting from the constraints it is given. It then branches out, working to compute a valid solution all while staying within the constraints imposed. Once a branch no longer fits within the constraints the algorithm backtracks and abandons that branch as a possible solution. However, instead of having to start again from scratch the algorithm backtracks to the point at which the constraints were lost allowing it to utilize all the work done to that point. This makes it much more efficient than brute force as it does not need to start from scratch each time. Backtracking is a common algorithm for solving crosswords, sudoku and other puzzles because due to their strict constraints allows the algorithm to work with high efficiency.

Backtracking is based on recursion and the spanning tree algorithm which are the basis for node by node searches in data structures. These protocols create the foundation for traversing a tree

and its nodes to search for a constraint variable. Backtracking works to build the tree backwards as instead of searching for the variable we are trying to complete a tree which will give us a valid solution.

*Figure 1*



## VI. EXPLANATION ON FILE I/O

We had also chosen to include the option for a user to modify a Comma Separated Values (CSV) file for both the user's convenience and so the program could be compiled into an executable file for easy access. The program uses the built in file reading system that is integrated into Python 3 and has an exception system based on the presence of input.csv. If the program doesn't find the file, it will throw an IO exception and will tell the user to create the file. If it is successful, it passes the values through the NumPy library and converts it from a string array into an integer array. There the program will run normally with the inputted array.

## REFERENCES

- [1] T. Altenkirch, Python Sudoku Solver - Computerphile, 21-Feb-2020.  
[Online]. Available: [https://www.youtube.com/watch?v=G\\_UYXzGuqvM](https://www.youtube.com/watch?v=G_UYXzGuqvM). [Accessed: 21-Nov-2020].
- [2] P. Youtube, Python CSV Files and 2D Arrays - How to Read and Process CSV Files into Python 2 Dimensional Arrays, 25-Mar-2020.  
[Online]. Available: <https://www.youtube.com/watch?v=qHRaVlCUr7Q>. [Accessed: 27-Nov-2020].