CS575: Final Project Report

**Project Title: Sudoku Solving Algorithm And Visulisation**

**Team Members:**

**Zimple Kabariya**

**Shivani Dhanasamy**

**Reshma Barvin Shahul Hameed Amanullah**

# Problem

The sole purpose of generating the Sudoku solving algorithm is to help the maker create a well-posed Sudoku grid. So our algorithm helps the person to check if a grid is solvable and if it has more than one solution.

Solving Sudoku has always been a time consuming task irrespective of its difficulty level. Using this algorithm developers or makers of the game can spend less time on developing or finding the solution of the game.

# Algorithms

## Naïve Algorithm: Elimination Algorithm

The naïve algorithm also known as elimination method is to generate all possible values to fill the empty cells.

If possibility set has only 1 value then it is entered in the grid else it move to next empty cell. We will iterate this till we find the correct placement for all cell. This is a very tedious process.

## Crook Algorithm

**Elimination**: a cell only has one possible value left.

**Lone rangers**: a number can only fit in one cell in a row or column or block.

## Backtracking

Assign possible value to the empty cells.

We begin traversing from the cell with fewest values in the possibility set.

But before doing so always check if that solution is safe or not. i.e. violating any constraints or not.

Repeat the above process recursively checking whether it leads to a solution or not after each assignment.

# Software Design and Implementation

We have designed our Sudoku using HTML, CSS and the algorithm is implemented using JavaScript.

Attachments

https://github.com/zkabari1/daa\_sudoku

##### References

[1]https://mpardesh.github.io/halide-bilateral-filtering-website/finalreport

[2]https://imgur.com/gallery/0sfWi

[3]http://brutalsimplicity.github.io/assets/sudoku/sudoku-backtrack.png

[4]https://hackernoon.com/sudoku-and-backtracking-6613d33229af