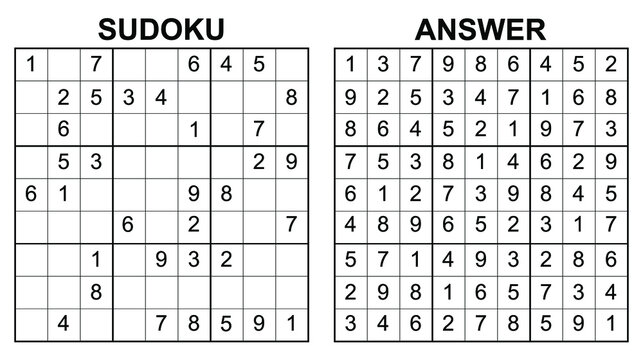
Sudoku Solver

Martin Rode, Marvin Üürike

Introduction

Sudoku is a popular number puzzle game that originated in Japan. The objective of the game is to fill a 9x9 grid with digits from 1 to 9 in such a way that each column, each row, and each of the nine 3x3 subgrids (also known as "boxes") contains all the digits from 1 to 9 without repetition. The puzzle starts with some cells pre-filled with numbers, and the player must fill in the remaining cells following these rules.

In this project, we aim to develop an AI Sudoku solver using the Depth-First Search (DFS) algorithm combined with various logical constraints. The goal is to enhance the solver's efficiency and accuracy in providing solutions to any valid Sudoku puzzle and to compare the effectiveness of different logical constraints in speeding up the solving process.



Strategies

1. Basic – insert numbers 1-9 into cells row by row and backtrack.
2. Storing legal values – Create a cache of legal values for each cell and only search through those. In this first version we do this only at the beginning, before solving
3. Updating cache – The most logical optimization is the one humans also use when solving sudokus. When we put a value into a cell we can remove it from the legal values of the cells in its row/cell/block. If we add searching through the cells with the lowest number of legal values first – this way we have a higher chance of getting to the answer more quickly,
4. Hidden singles – If it happens that a value can only legally be placed in one cell, we can safely put that value into the cell (even if the cell has multiple legal values beforehand).
5. Priority number search – When trying the legal values of a cell, we can start with the ones that occur the least in the legal values of the other cells in its row/cell/block