

Sudoku Solver

Introduction A Sudoku solver is a program designed to solve Sudoku puzzles by filling a 9x9 grid with digits 1 through 9. Each row, column, and 3x3 subgrid must contain all digits exactly once. This document explains the data structures and algorithms used to build a Sudoku solver.

1. Data Structures Used

1. Grid Representation:

- **Data Structure:** 2D Array (List of Lists in Python).
- **Purpose:** Represents the Sudoku board where each cell holds a number (1-9) or 0 for empty cells.

2. Sets for Validation:

- **Data Structure:** Sets.
- **Purpose:** Tracks digits in rows, columns, and 3x3 subgrids efficiently.

3. Recursive Stack:

- **Purpose:** Implements backtracking by storing the board's state during solving.
-

2. Algorithm Explanation

The solver uses **Backtracking**, a trial-and-error approach to fill the grid:

1. Find the Next Empty Cell:

- Locate the first empty cell (0).

2. Try a Number:

- Place a number (1-9) and validate:
 - Ensure the number isn't in the same row, column, or 3x3 box.

3. Recursive Call:

- Move to the next empty cell and repeat.

4. Backtrack:

- Undo the last number if no valid number fits.

5. Terminate:

- Stop when all cells are filled or the puzzle is unsolvable.
-

3. Key Functions

1. **is_valid(number, row, col, grid):**

- Checks if a number placement is valid in a specific cell.

2. **solve(grid):**

- Implements the backtracking algorithm to solve the puzzle.
-

4. Advantages

1. **2D Array:** Simple and efficient for grid representation.
 2. **Sets:** Fast lookups for validation, avoiding redundant checks.
 3. **Recursive Stack:** Naturally handles backtracking.
-

5. Limitations

1. Backtracking can be slow for complex puzzles.
 2. Recursive calls require sufficient memory.
-

Screenshots:

Sudoku Solver

		8	1			7		6
	2	7			6			8
		6		8		5	9	2
	7	9	2	4	8			
					3	8		
8				9		3	7	
	8				2			
				6		2		3
2		1					4	7

Solve

Clear

Enter a Sudoku puzzle to solve

Sudoku Solver

9	2	8	1	5	4	7	3	6
5	3	7	9	2	6	4	1	8
1	4	6	3	8	7	5	9	2
3	7	9	2	4	8	1	6	5
4	6	5	7	1	3	8	2	9
8	1	2	6	9	5	3	7	4
6	8	3	4	7	2	9	5	1
7	9	4	5	6	1	2	8	3
2	5	1	8	3	9	6	4	7

Solve

Clear

Enter a Sudoku puzzle to solve

Conclusion: The Sudoku solver combines a 2D array, sets, and a backtracking algorithm to solve puzzles. The use of appropriate data structures ensures clarity and performance.