

REPORT FILE

**Problem statement: Sudoku
Solver**

Name: Mahima Bhat

Roll No: 202401100400116

Course: Introduction to AI

Date: 10-03-2025

Introduction-

Sudoku is a popular logic-based number puzzle that requires filling a 9x9 grid so that each column, each row, and each of the nine 3x3 grids contain all digits from 1 to 9. The objective of this project is to develop a Python program that can efficiently solve any given Sudoku puzzle using the backtracking algorithm.

The provided Sudoku Solver takes an incomplete Sudoku board as input and fills in the missing numbers while ensuring all constraints are satisfied. The backtracking algorithm explores possible solutions by testing each potential number and backtracking when conflicts arise.



Methodology-

The Sudoku Solver is implemented using the following steps:

1. Input the Puzzle: The unsolved Sudoku puzzle is provided as a 9x9 matrix where empty cells are marked as zero.

2. Validation Check: The program verifies if placing a particular number in a cell is valid by checking:

- Row
- Column
- 3x3 Grid

3. Backtracking Algorithm:

- The algorithm iterates through each cell in the puzzle.
- If an empty cell is encountered, the program attempts to place numbers from 1 to 9.
- If a valid number is found, it proceeds further; otherwise, it backtracks and tries alternative numbers.

4. Output the Solution: The solved Sudoku puzzle is displayed in a clear format.

```
5. def is_valid(board, row, col, num):
6.     # Check if the number is not in the current row and column
7.     for i in range(9):
8.         if board[row][i] == num or board[i][col] == num:
9.             return False
10.
11.     # Check if the number is not in the 3x3 box
12.     box_start_row, box_start_col = 3 * (row // 3), 3 * (col
// 3)
13.     for i in range(3):
14.         for j in range(3):
15.             if board[box_start_row + i][box_start_col + j]
== num:
16.                 return False
17.
18.     return True
19.
20. def solve_sudoku(board):
21.     for row in range(9):
22.         for col in range(9):
23.             if board[row][col] == 0: # Empty cell found
24.                 for num in range(1, 10): # Try numbers 1 to
9
25.                     if is_valid(board, row, col, num):
26.                         board[row][col] = num
27.
28.                     if solve_sudoku(board): #
Recursively attempt to solve
29.                         return True
30.
31.                     # Backtrack if the current number
doesn't work
32.                     board[row][col] = 0
33.                     return False # Trigger backtracking
34.         return True
35.
36. def print_sudoku(board):
37.     for row in board:
38.         print(" ".join(str(num) if num != 0 else '.' for num
in row))
39.
40.     # Sample Sudoku Puzzle (0 represents empty spaces)
41.     sudoku_board = [
42.         [5, 3, 0, 0, 7, 0, 0, 0, 0],
```

```

43.         [6, 0, 0, 1, 9, 5, 0, 0, 0],
44.         [0, 9, 8, 0, 0, 0, 0, 6, 0],
45.         [8, 0, 0, 0, 6, 0, 0, 0, 3],
46.         [4, 0, 0, 8, 0, 3, 0, 0, 1],
47.         [7, 0, 0, 0, 2, 0, 0, 0, 6],
48.         [0, 6, 0, 0, 0, 0, 2, 8, 0],
49.         [0, 0, 0, 4, 1, 9, 0, 0, 5],
50.         [0, 0, 0, 0, 8, 0, 0, 7, 9]
51.     ]
52.
53.     print("Original Sudoku Board:")
54.     print_sudoku(sudoku_board)
55.
56.     if solve_sudoku(sudoku_board):
57.         print("\nSolved Sudoku Board:")
58.         print_sudoku(sudoku_board)
59.     else:
60.         print("\nNo solution exists for the given Sudoku
61.         board.")

```

The screenshot shows a web browser with several tabs, including 'Sudoku Solver Code'. The active page is a Jupyter Notebook titled 'MahimaBhat_202401100400116.ipynb'. The notebook interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for commands, code, and text, and a main area displaying the output of the code. The output shows the 'Original Sudoku Board' and the 'Solved Sudoku Board'. The status bar at the bottom indicates '0s completed at 2:19 PM'.

```

Original Sudoku Board:
5 3 . . 7 . . .
6 . . 1 9 5 . . .
. 9 8 . . . . 6 .
8 . . . 6 . . . 3
4 . . 8 . 3 . . 1
7 . . . 2 . . . 6
. 6 . . . . 2 8 .
. . . 4 1 9 . . 5
. . . . 8 . . 7 9

Solved Sudoku Board:
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9

```

References/Credits for images-

[https://d2.alternativeto.net/dist/s/sudoku-solver-and-](https://d2.alternativeto.net/dist/s/sudoku-solver-and-generator_214277_full.png?format=jpg&width=1200&height=1200&mode=crop)

[generator_214277_full.png?format=jpg&width=1200
&height=1200&mode=crop](https://d2.alternativeto.net/dist/s/sudoku-solver-and-generator_214277_full.png?format=jpg&width=1200&height=1200&mode=crop)