# Sudoku Solver Report

## Title Page

Problem Statement: Develop a Python program to solve a given Sudoku puzzle using the backtracking algorithm.

Name: Ansh Rajput

Roll No : 202401100300052

Date : 11/3/25

Instructor : Mr. Bikki Kumar

## Introduction

Sudoku is a logic-based number puzzle that requires filling a 9×9 grid such that each row, each column, and each of the nine 3×3 subgrids contain all digits from 1 to 9 without repetition. The given problem involves implementing a Python program that can automatically solve any Sudoku puzzle using the backtracking algorithm.

## Methodology

1. Finding Empty Cells: Identifies the first unfilled cell.

2. Checking Validity: Ensures numbers follow Sudoku rules.

3. Recursive Backtracking: Places numbers recursively.

4. Backtracking: If stuck, removes last number and tries another.

5. Completion: Continues until the grid is fully solved.

## Code

def is\_valid(board, row, col, num):  
 for i in range(9):  
 if board[row][i] == num or board[i][col] == num:  
 return False  
 start\_row, start\_col = 3 \* (row // 3), 3 \* (col // 3)  
 for i in range(3):  
 for j in range(3):  
 if board[start\_row + i][start\_col + j] == num:  
 return False  
 return True  
  
def find\_empty\_cell(board):  
 for i in range(9):  
 for j in range(9):  
 if board[i][j] == 0:  
 return i, j  
 return None  
  
def solve\_sudoku(board):  
 empty = find\_empty\_cell(board)  
 if not empty:  
 return True  
 row, col = empty  
 for num in range(1, 10):  
 if is\_valid(board, row, col, num):  
 board[row][col] = num  
 if solve\_sudoku(board):  
 return True  
 board[row][col] = 0  
 return False

## Output/Result

The solved Sudoku grid is as follows:  
(Screenshot of the solved Sudoku puzzle is attached below)