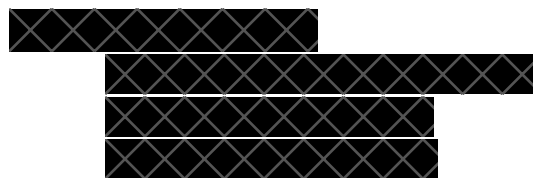


# SUDOKU

CS 102

DATA STRUCTURES & ALGORITHMS  
PROJECT PROPOSAL



## Description :

Sudoku is a logic based puzzle game, to win the player has to place a number in a way that the number is not repeated in the same column or row or 3x3 subgrid and the numbers used can be from 1-9 therefore it is a 9x9 grid overall.

We will use backtracking algorithm to generate a Sudoku board, first we will create an empty 9x9 matrix then we will input 1 random valid value in a random position on the board and use the backtracking algorithm to produce a unique Sudoku board, which we will then solve on our own and then check for answers.

Manually solving this logic based puzzle can take a lot of time so we will use a backtracking algorithm to make the process faster.

## Valid Moves (Constraints):

Input any number that is not present in the Row, Column or Sub-Grid

## Goal:

The game will be considered successful/won if no empty space is left on the board.

## Algorithm Used:

We will be implementing the backtracking algorithm. The player will input values 1-9 in such a manner that no value is present in the row, column, or sub-grid. These values will also be placed using the backtracking and random number initially given. He will repeat this process until he reaches a point where no such solution is possible, this is when backtracking will take place. The player will move back to his previous solution and change it. If, for example, at the previous position, no such solution is possible again (aside from the solution he removed), he will take another step back and remove that solution and try another. He will repeat this process until he finds a valid solution and then move forward.

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

The grid above is not complete and is for understanding purposes only. The highlighted numbers are the solutions that a player has entered. The solution highlighted in blue indicates that it is an invalid solution as the same number is present in that column. And so, the game will backtrack and go back to the previous solution highlighted in green, and the player will try another solution for that position.

#### Project Outcome:

We will be using a backtracking algorithm to make a sudoku game. We are using this algorithm so that all possibilities of players number placements can be checked and all combinations can be checked. To reduce the overall runtime and produce a valid outcome.

#### Libraries/other resources to be used:

We will be using pyGame to implement the interface of our game. We will be using the random module to create a game board that has random values each time the game is started in non-empty boxes (an empty box is indicated by a 0) (this is tentative, using random values does not ensure that the game is solvable.)