

SUDOKU GAME

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

SUDOKU GAME

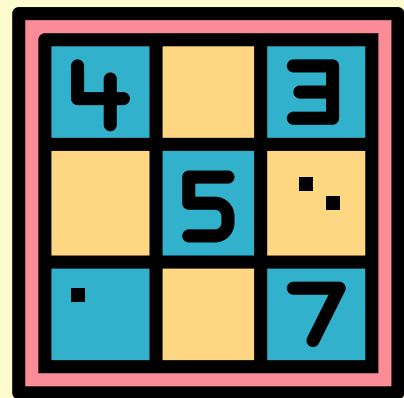
backtracking Approach

naive Approach

sudoku game problem

Given a partially filled 9×9 2D array 'grid[9][9]'.

the goal is to assign digits(from 1 to 9) to the empty cells so that every row, column, and subgrid of size 3×3 contains exactly one instance of the digits from 1 to 9.



backtracking

A backtracking algorithm is a recursive algorithm that attempts to solve a given problem by testing all possible paths towards a solution until a solution is found. Each time a path is tested, if a solution is not found, the algorithm backtracks to test another possible path and so on till a solution is found or all paths have been tested.

naive

naive Algo generate all possible configurations. Try every configuration one by one until the correct configuration is found. for every unassigned position fill the position with specific configuration . After filling all the unassigned position check if it's safe or not. If safe print else recurs for other cases.

backtracking Approach to solve sudoku game

`def solveSudoku():`

this is recursive function we used it to solve the sudoku game and This function calls other functions to help this function solve.

`def findNextCellToFill():`

This function is used to search for empty cells.

`def isValid():`

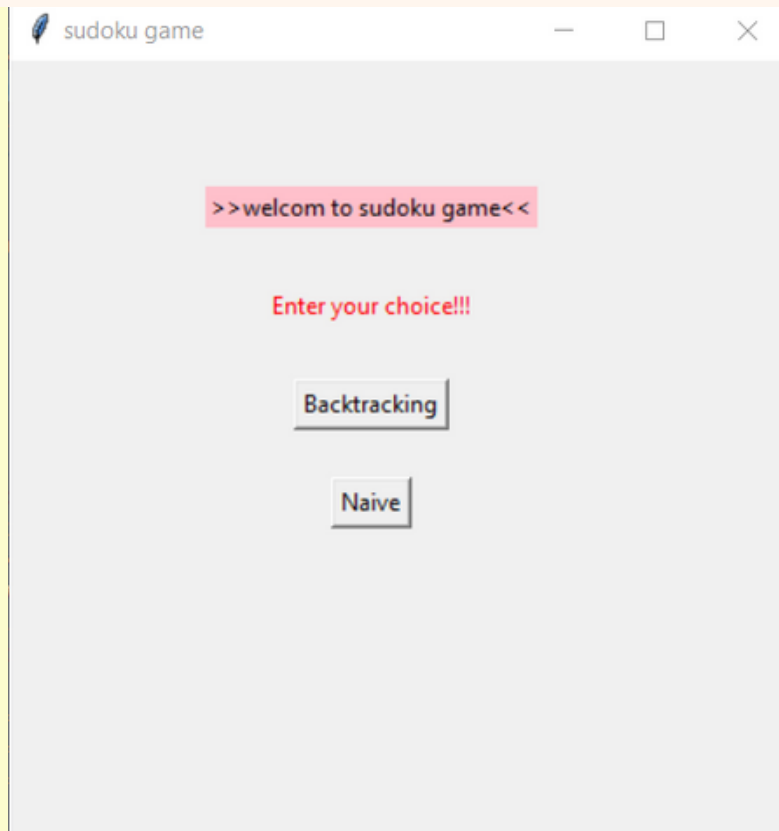
This function is used to test whether the numbers in the cells are valid or not.

`def printsudoku():`

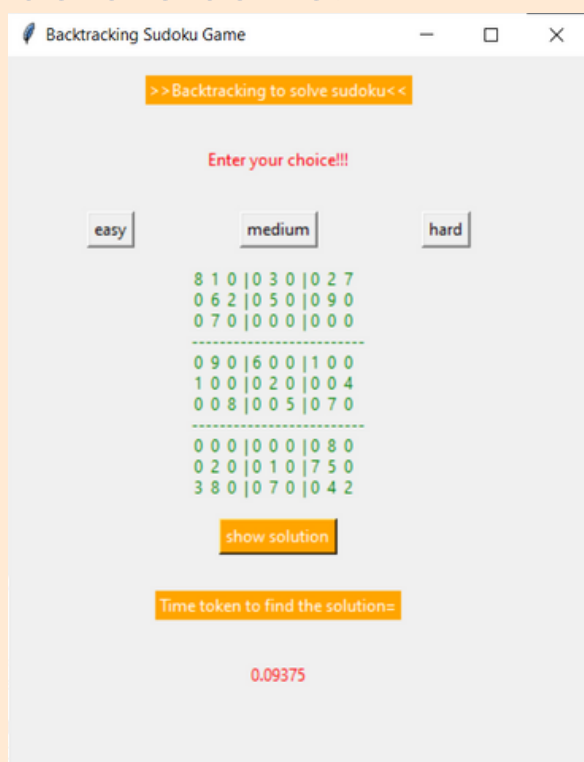
This function is used to print Sudoku before and after the solution .

we use the tkinter library to implement the animation.

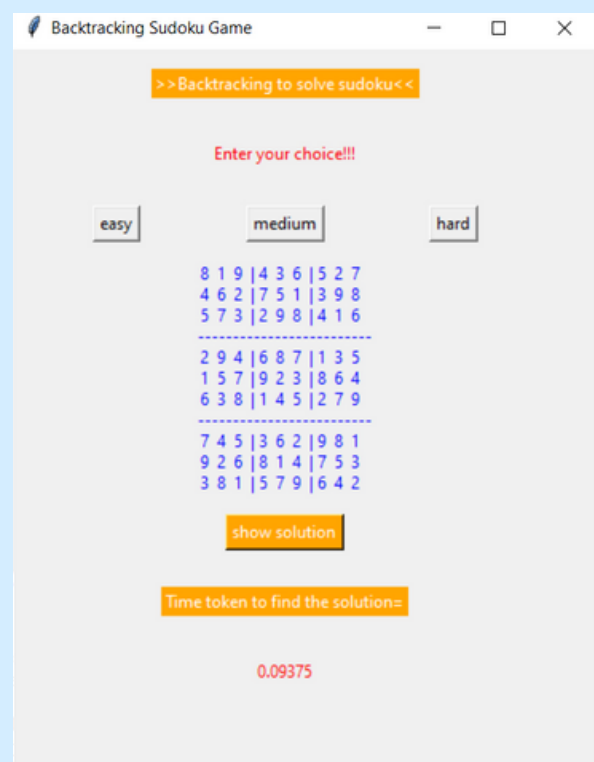
backtracking Approach example



before solve:



after solve:



naive Approach to solve sudoku game

`def solveSudoku():`

this is recursive function we used it to solve the sudoku game and This function calls other functions to help this function solve.

`def isSafe():`

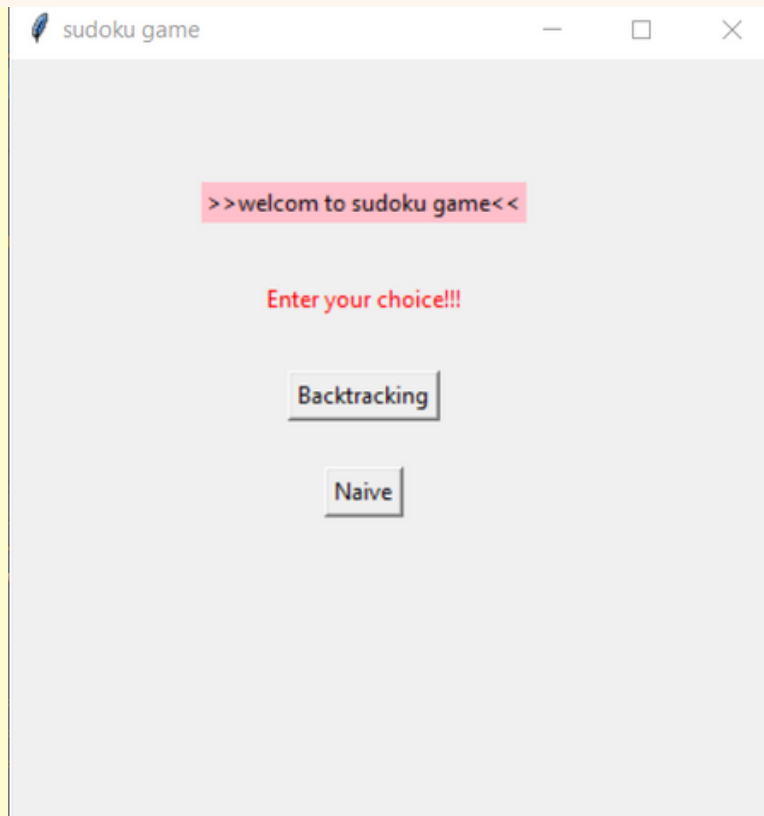
this function checks if the given matrix is valid sudoku or not.

`def printsudoku():`

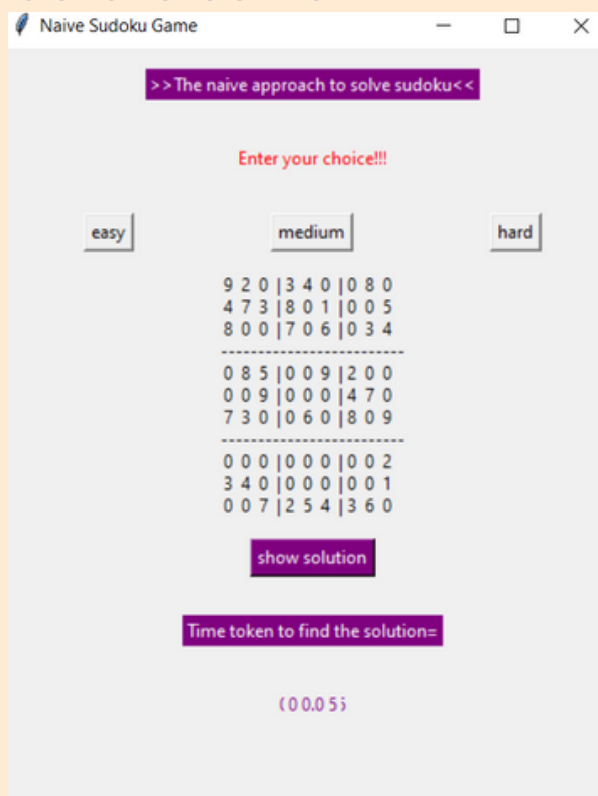
This function is used to print Sudoku before and after the solution .

we use the tkinter library to implement the animation.

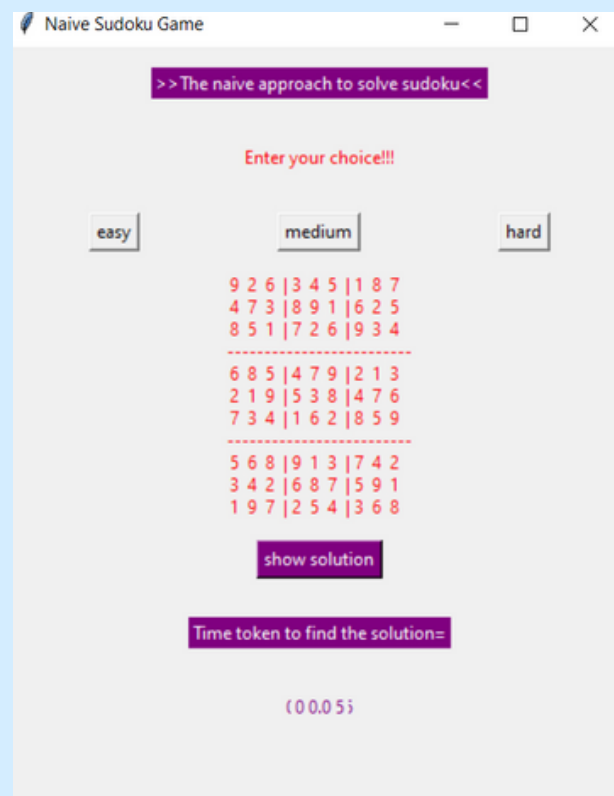
naive Approach example



before solve:



after solve:



Comparison between algorithms

Time complexity

backtracking

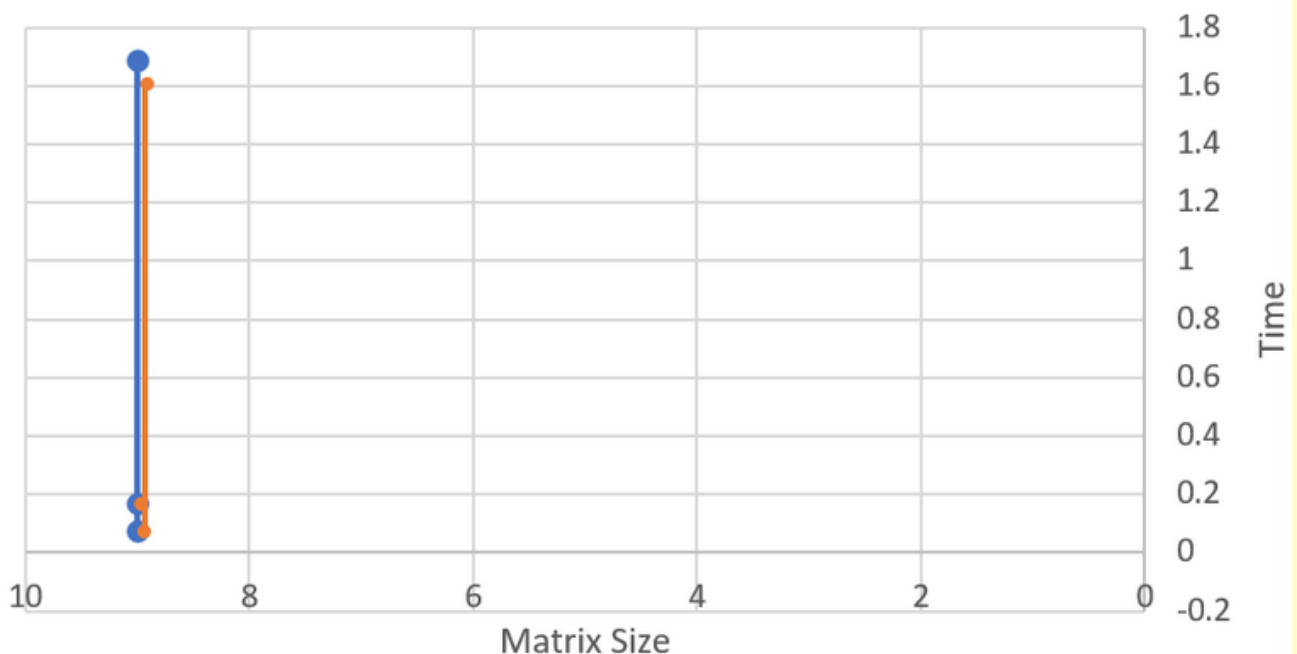
$O(9^{(n*n)})$ for every unassigned index, there are 9 possible options so the time complexity is $O(9^{(n*n)})$.

naive

$O(9^{(n*n)})$ for every unassigned index, there are 9 possible options so the time complexity is $O(9^{(n*n)})$.

The time complexity for both are the same, but there will be some early pruning so the time taken in **backtracking Algorithm will be much less than the naive algorithm** but the upper bound time complexity remains the same.

Time Comparison



Group members

Alaa Hasan Said

ID:441016502

Safa Hasan Said

ID:441016546

talla Ahmed

ID:441015910

raghad yousef

ID:441005529

