Logic behind the code

1. Constraint Checking
Before diving into DFS and Backtracking, it's crucial to understand constraint checking:

Constraints in Sudoku are:
Each number 1-9 must appear exactly once in each row.
Each number 1-9 must appear exactly once in each column.
Each number 1-9 must appear exactly once in each 3x3 sub-grid.
Constraint Checking ensures that a number placed in a cell doesn't violate these rules.

Steps:

For any number being considered for a cell, check if it’s already present in the current row, column, or 3x3 grid.
If the number doesn't violate any constraints, it can be placed in the cell.

****

2. Depth-First Search (DFS)
DFS is a search algorithm that explores as far as possible along each branch before backing up. In Sudoku solving:

DFS is used to explore potential solutions by placing numbers in empty cells.
DFS starts with an empty cell, tries placing each number (1-9), and recursively proceeds to solve the rest of the board.
DFS continues to explore possible numbers for subsequent empty cells until it finds a solution or exhausts all possibilities.
Steps:

Find an empty cell in the grid.
For each possible number (1-9), check if placing the number in the empty cell is valid.
Recursively apply DFS to the next empty cell.
If placing a number leads to a solution, return true.
If no number fits and leads to a solution, backtrack (undo the number placement) and try the next number.

****

3. Backtracking
Backtracking is a refinement of DFS. It involves:

Trying a potential solution (placing a number in an empty cell).
Checking if it leads to a valid solution using DFS.
If it leads to a valid solution, keep it.
If it doesn't, undo the move (remove the number) and try another number.

$$$

Steps:

After placing a number in an empty cell, attempt to solve the puzzle with this new configuration.
If a number doesn’t lead to a solution, reset the cell (backtrack) and try the next number.
Combining DFS, Backtracking, and Constraint Checking
Initialization:

Start with the current board configuration.

Identify the first empty cell.
Constraint Checking:

For each empty cell, try placing numbers 1-9.
Use constraint checking to ensure the number doesn’t violate Sudoku rules.
DFS Exploration:

Place a number in the empty cell.
Use DFS to solve the remaining board configuration.
If DFS finds a solution, return true.
Backtracking:

If DFS cannot solve the board with the current number, reset the cell (remove the number) and try the next number.
Completion:

Repeat until the board is completely and correctly filled or all possibilities are exhausted.