

Assignment 4

Agbola Iseoluwatobi B00802526

CSCI 3901

Overview

The Program Implements a sudoku solver. The program asks for the size of the grid which must be a perfect square integer and then the program requests for the list of characters in the grid. Then a string of the square of the grid size is inputted. After the inputs are validated a grid of objects is created and then the grid is passed to a solver object that attempts to solve the grid. The grid is solved with a back tracking recursive algorithm that uses brute force to solve the sudoku puzzle by trying all possible valid inputs until we find the right sequence of valid inputs.

File and External data

The main files correspond to the classes in the game.

Main.java	This is the main class where all the objects are called, and the input are accepted and validated.
Grid.java	This is the class of the grid object it stores a multi-dimensional of type sudokuElement represents the game. The object of this class is passed a solver object
Solver.java	This class takes in a grid object and solves the puzzle with a back tracking recursive algorithm. It contains validators to detect when a grid is fully solved
SudokuElement.java	The objects of the class represent each cell in the grid. It stores the value of each cell.

Data structures and their relations to each other's

Grid object: this stores the multi- dimensional array grid of sudokuElement objects. It also stores the integer gridSize and a set gridCharacters

SudokuElement object: this object stores the value of each sudoku cell in a String value.

Solver object: takes in the Grid Object and solves the puzzle

Grid multi-dimensional array: store all the elements of the puzzle of type sudokuElement

Integer gridSize: store the size of the grid.

Set gridCharacters: store a set of the characters in the grid

Assumptions

1. The size of the grid is a perfect square
2. The number of unique characters is equal to the side of the grid
3. The sudoku puzzle has a solution
4. The solution is not the worst-case scenario for the recursive backtracking algorithm
5. The number of characters in the input string is equal to number of elements in the grid

Key algorithms and design elements

Recursive backtracking: This is an algorithm that used to solve computational problems it involves exploring possible options and backtracking the moment an option is not possible or invalid. It is a form of brute force algorithm and is useful for state space exploration problems.

Limitations

1. The program slows down as the grid size increases and the when the elements are arranged the worst-case scenario for a brute force algorithm.
2. The grid size must be a perfect square.

Test Cases

- The user enters a grid size that is a perfect square
- The user enters a grid size that is not a perfect square
- The user enters a negative number as grid size
- The user enters a positive number as grid size
- The user enters zero as the grid size
- The user enters a character as a grid size
- The user enters the number of characters that is equal to the grid size
- The user enters the number of characters that are not equal to the grid size
- The character list entered by the user are all unique characters
- The characters list entered by the user are not unique
- The sudoku puzzle entered is a valid puzzle
- The sudoku puzzle entered is not a valid puzzle
- The user enters a puzzle of a low grid size
- The user enters a puzzle of a high grid size

References

Gurari, Eitan (1999). "CIS 680: DATA STRUCTURES: Chapter 19: Backtracking Algorithms"

