Sudoku Evaluation Document

**Introduction**

The aim of this report is to outline the the design and implementation of my take on a command-line version of Sudoku using the C programming language. The primary objectives included efficiently representing and manipulating the Sudoku game board, implementing additional features such as difficulty settings, game recording’s and a progress bar.

**Critical Evaluation:**

1. Data Structures and Algorithms.

* Efficient Use of 2-D Arrays: Utilizing a 2-dimensional array in C to represent the game grid is a suitable choice for simplicity and ease of access.
* Recursive Algorithm for Generating Sudoku Grid: The recursive algorithm for populating the initial Sudoku Grid ensures a valid game state. However, the efficiency of this algorithm could be further evaluated for if larger grids are introduced to the game.
* Struct for player representation: Using a struct to store necessary data related to the current player helps simplify access and manipulation of game elements, providing better organization.
* File Storage: Storing game data in a JSON file allowed for efficient data management and persistence between sessions through the use of a user-specific ID

1. Additional Features:

* Difficulty Settings: Providing options for different difficulty levels enhances the games replay ability and caters to players with varying skill levels.
* Game Recording: These features significantly improve the user experience by allowing players to review and continue play of games previously unfinished, by Giving the player their specific ID at the end of session it allows some security in how their game is stored.
* Progress Bar: The visual representation of the progress bar provides visual feedback to the player, enhancing their overall gaming experience.

1. Unimplemented features:

* Scoring and Leaderboard Functionality: Given more time, I would continue to implement a scoring system to add a more competitive aspect to the game, providing difficulty modifiers and bonuses based on time taken to complete.
* Undo/Redo mechanics: To provide my Sudoku with more QoL features as well as enhance the user experience, Undo/Redo mechanics would have been implemented where appropriate and ensure smooth functionality.

1. Experimental Results

* Performance Testing: Conducting performance tests, especially for the recursive algorithm used in grid generation, would provide valuable insights into the efficiency of implementation
* User Feedback: Gathering feedback from users on the useability and effectiveness of the additional features would help identify areas for improvement and refinement
* Timing: Setting up clocks throughout the program would help identify areas of the code that are less effective and allow action to be taken to address that

**References:**

Modified code from Geeks for Geeks to read and write my Json File

<https://www.geeksforgeeks.org/cjson-json-file-write-read-modify-in-c/>

Used the ultralightweight JSON parser

[https://github.com/DaveGamble/cJSON?tab=readme-ov-file#usage](https://github.com/DaveGamble/cJSON?tab=readme-ov-file%23usage)