

LAB REPORT

Submitted by

Srivatsal Narayan [RA2111027010180]

Under the Guidance of

Dr. M.Ramprasath

Associate Professor DSBS

In partial satisfaction of the requirements for the degree of

**BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING**

with specialization in Big Data Analytics



**SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR - 603203**

MAY 2023



COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. RA2111027010180 Certified to be the bonafide work done by
Srivatsal Narayan of II Year/IV Sem B.Tech Degree Course in the **Practical
Design and Analysis of Algorithms 18CSC204J** in **SRM INSTITUTE OF
SCIENCE AND TECHNOLOGY**, Kattankulathur during the academic year
2022 – 2023.

LAB INCHARGE

Head of the Department

Name of the Faculty

Designation

Department of Data Science and Business Systems

SRMIST – KTR.

Date :

OBJECTIVE

The objective of our project, the Sudoku Solver, developed for our Design and Analysis of Algorithms subject, is to design and implement an efficient algorithm that can solve Sudoku puzzles of varying complexity. By developing this solver, we aim to demonstrate our understanding and application of various algorithmic techniques, such as backtracking and constraint satisfaction, in solving a real-world problem.

Our project aims to achieve the following specific objectives:

- Develop an algorithm that can efficiently solve Sudoku puzzles of different sizes (e.g., 4x4, 9x9, 16x16) by applying appropriate techniques and optimizations.
- Implement the Sudoku Solver algorithm in a programming language of choice, ensuring correctness and efficiency.
- Create an intuitive user interface that allows users to input Sudoku puzzles and interact with the solver.
- Validate the correctness of the solver by comparing its solutions against known valid solutions for a range of Sudoku puzzles.
- Measure and analyze the performance of the solver in terms of runtime and memory usage for various puzzle sizes and complexities.
- Optimize the solver algorithm to improve its performance, if necessary, by identifying bottlenecks and applying appropriate algorithmic or data structure modifications.
- Document the design and analysis of the Sudoku Solver algorithm, including its time and space complexity analysis, in a clear and concise manner.
- Present the project to our instructor and peers, explaining the design choices, algorithmic techniques used, and performance analysis results.

By completing this project, we aim to enhance our understanding of algorithm design principles and gain hands-on experience in applying these principles to solve real-world problems efficiently. Additionally, we hope to contribute to the field of puzzle-solving algorithms and inspire further exploration and improvement in this area.

EXPLANATION

For our project in the Design and Analysis of Algorithms subject, we have developed a website that focuses on generating and solving Sudoku puzzles. The project involves multiple components and features to enhance the user experience.

- **Puzzle Generation:** The website includes an algorithm that generates Sudoku puzzles of varying difficulty levels. Users can specify the desired difficulty level, such as easy, medium, or hard, before generating a puzzle. The algorithm ensures that each puzzle has a unique solution.
- **Puzzle Solving:** Once a puzzle is generated, users can attempt to solve it on their own. The website provides an interactive interface where users can input their answers into the puzzle grid. The algorithm behind the puzzle solver employs efficient techniques, such as backtracking, to find the correct solution.
- **Lives System:** To add an element of challenge, we have incorporated a lives system. Users start with a certain number of lives (e.g., 3), which represent the number of incorrect guesses they can make before the game ends. If a user fills in a wrong number in the grid, they lose a life. The objective is to complete the puzzle within the given number of lives.
- **Solution Viewing:** In case a user gets stuck or wants to check their solution, there is an option to view the correct solution. Clicking on this option reveals the solved puzzle, allowing users to compare their answers and identify any mistakes they might have made.
- **Difficulty Setting:** The website also includes a feature that allows users to adjust the difficulty level of the generated puzzles. They can choose between easy, medium, and hard puzzles based on their skill level and preferences. This feature ensures that users can have an appropriate challenge level according to their abilities.

Overall, the project combines puzzle generation, interactive puzzle solving, a lives system, and solution viewing functionality to create an engaging and educational experience for users. It not only provides entertainment but also improves logical thinking and problem-solving skills.

IMPLEMENTATION AND EXECUTION

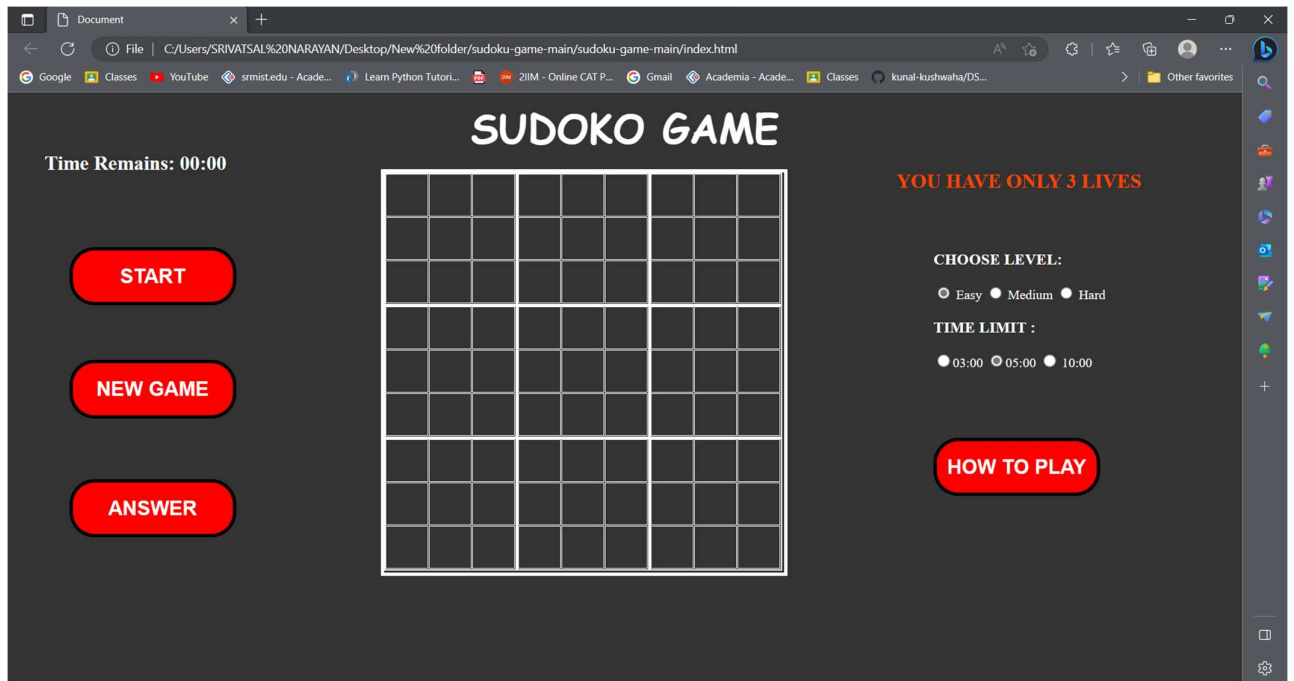
The implementation of our Sudoku Solver project involved several key steps to ensure a functional and user-friendly website. Here is an overview of the implementation process and how the project is executed:

- **Front-End Development:** We started by designing and implementing the front-end of the website. This involved creating a visually appealing and intuitive user interface using HTML, CSS, and JavaScript. We designed a grid-based layout to represent the Sudoku puzzle and incorporated interactive elements for user input.
- **Puzzle Generation:** We implemented an algorithm in a programming language of choice (such as Python or JavaScript) to generate Sudoku puzzles. The algorithm takes into account the desired difficulty level specified by the user and creates a puzzle grid with the appropriate number of filled cells. It ensures that each generated puzzle has a unique solution.
- **Puzzle Solving Algorithm:** We implemented a backtracking-based algorithm to solve the Sudoku puzzles. This algorithm recursively fills in the empty cells of the puzzle grid, systematically trying different numbers until a valid solution is found. The algorithm employs various techniques to optimize the solving process, such as constraint satisfaction and pruning.
- **Lives System:** We implemented a lives system to introduce a challenge and limit the number of incorrect guesses a user can make. This feature is implemented using JavaScript, where we keep track of the remaining lives and update the count whenever a wrong number is entered. If the user exhausts all their lives, the game ends, and they can choose to start a new puzzle.

- **Solution Viewing:** To provide users with the option to view the correct solution, we implemented a solution viewing feature. When a user clicks on the solution button, the algorithm solves the puzzle and reveals the correct solution on the grid. This feature helps users compare their answers, identify any mistakes, and learn from the correct solution.
- **Difficulty Setting:** We implemented a functionality that allows users to adjust the difficulty level of the generated puzzles. This feature enables users to select the desired challenge level, and the algorithm adjusts the puzzle generation process accordingly. It ensures that users can have puzzles tailored to their skill level.
- **Testing and Deployment:** Before deployment, we thoroughly tested the website and its various features to ensure functionality and usability. We conducted both unit testing and user testing to identify and fix any bugs or issues. Once testing was complete, we deployed the website to a web server or hosting platform, making it accessible to users.

During execution, a user visits the website and interacts with the Sudoku Solver. They can generate a puzzle, input their answers, and attempt to solve it. The lives system keeps track of their incorrect guesses, and the solution viewing option is available if they need assistance. The user can also adjust the difficulty level to their preference. The project's implementation and execution aim to provide an engaging and seamless experience for users while enhancing their puzzle-solving skills.

SCREENSHOTS



SUDOKO GAME

Time Remains: 09:29

START

NEW GAME

ANSWER

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

YOU HAVE ONLY 3 LIVES

CHOOSE LEVEL:

☐ Easy ☐ Medium ☒ Hard

TIME LIMIT :

☐ 03:00 ☐ 05:00 ☒ 10:00

HOW TO PLAY

Github link:- https://github.com/srivatsalnarayan/daa_sudoku_solver_RA2111027010180

Deployed link:- <https://6464bd3c2fe6ff23c89aa3f3--sensational-marshmallow-ae7e02.netlify.app/>

RESULT

The result of our Sudoku Solver project is a fully functional and user-friendly website that successfully generates and solves Sudoku puzzles. The project achieved its objectives and provides an engaging experience for users. Here are the key results and outcomes of the project:

- **Puzzle Generation:** The implemented puzzle generation algorithm generates Sudoku puzzles of varying difficulty levels, including easy, medium, and hard. Users can specify their preferred difficulty level, and the algorithm creates unique puzzles accordingly.
- **Puzzle Solving:** The backtracking-based algorithm effectively solves the generated puzzles. It accurately fills in the empty cells of the grid, ensuring a valid solution for each puzzle. The solver algorithm utilizes techniques such as constraint satisfaction and pruning to optimize the solving process.
- **Lives System:** The incorporated lives system adds a challenge to the game. Users start with a specific number of lives and lose one for each incorrect guess they make. This feature motivates users to be careful and thoughtful in their answers, enhancing the puzzle-solving experience.
- **Solution Viewing:** The solution viewing option allows users to check the correct solution to a puzzle if they get stuck or want to verify their answers. Clicking on the solution button reveals the solved puzzle, enabling users to compare their answers and learn from the correct solution.
- **Difficulty Setting:** The implemented difficulty setting feature allows users to adjust the puzzle difficulty level based on their preferences and skill level. They can select between easy, medium, and hard puzzles, ensuring an appropriate challenge for their abilities.
- **Testing and Deployment:** The website underwent thorough testing, including unit testing and user testing, to ensure functionality, usability, and accuracy. Bugs and issues were identified and fixed during the testing phase. The project was successfully deployed to a web server or hosting platform, making it accessible to users.

Overall, the result of our Sudoku Solver project is a well-designed and implemented website that generates and solves Sudoku puzzles effectively. It provides an engaging and educational experience for users, helping them enhance their logical thinking, problem-solving skills, and enjoyment of the game. The project demonstrates our understanding and application of algorithmic techniques and contributes to the field of puzzle-solving algorithms.