N-Queens Multi-Threaded Solution Documentation

Project Description:

The N-Queens Multi-Thread GUI is a Java Swing application designed to solve and visualize solutions for the N-Queens problem. This classic chess puzzle requires placing N queens on an N×N chessboard in a way that avoids mutual threats.

Implementation Overview:

Graphical User Interface (GUI):

- **Main Frame:**
  - The application uses Java Swing to create a user-friendly interface.
  - The main frame includes input fields, buttons, and an output text area.

N-Queens Solver:

- **Multi-Threaded Approach:**
  - The application employs an ExecutorService to manage multiple threads concurrently.
  - The number of threads is dynamically adjusted based on user input and available processor cores.

- **Dynamic Thread Management:**
  - The user specifies the number of queens via an input field.
  - The application intelligently determines the optimal number of threads for performance.

Code Documentation:

Main Method (main):

- Initializes the Swing GUI components.
- Listens for user input to determine the number of queens and starts the solution generation process.

solveNQueens Method:

- Manages the execution of multiple threads using an ExecutorService.
- Dynamically adjusts the number of threads based on user input and available processor cores.

solveNQueens (Overloaded) and placeQueens Methods:

- Implements the core logic for solving the N-Queens problem using backtracking.
- Places queens on the chessboard and validates their positions.

isValidPlacement Method:

- Checks whether placing a queen at a specific position is valid, considering previously placed queens.

printSolution Method:

- Converts the chessboard configuration into a readable string format.
- Appends the solution to the Swing GUI's text area.

Algorithm Used:

The application employs a backtracking algorithm to explore and find solutions to the N-Queens problem. This algorithm recursively places queens on the chessboard, checking the validity of each placement.

Additional Information:

- The number of threads is dynamically adjusted based on user input and the available processor cores for optimal performance.
- The GUI provides real-time updates on found solutions.

Member Rule:

1. Mohamed Sroor: Final integration and testing
2. Radwa Mostafa: GUI setup and documentation
3. Eman Mohamed: N-Queen algorithm
4. Mohamed Abd-Elahady: Input validation
5. Ammar Yasser: Thread management
6. Salma Ahmed: Printing solutions