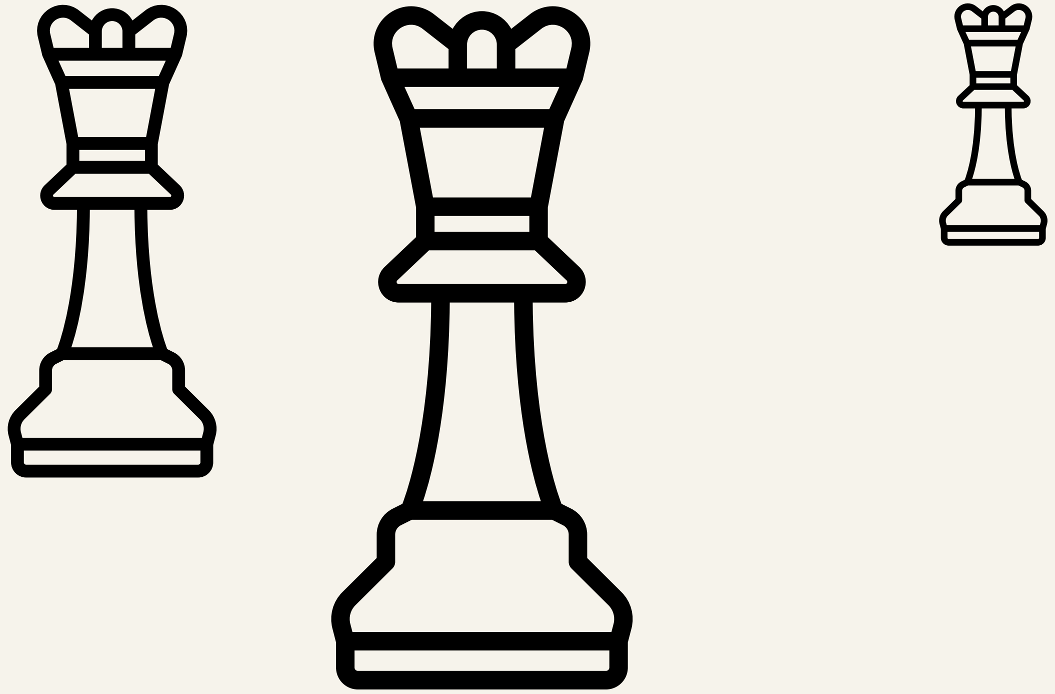
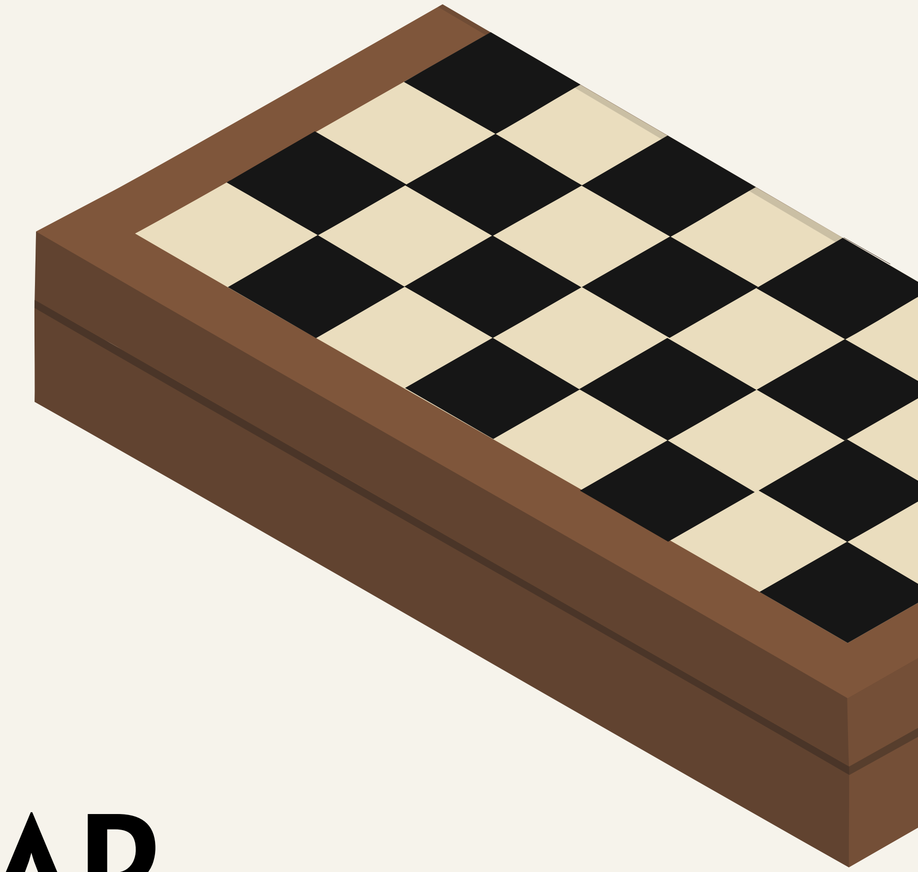


N QUEENS PROBLEM

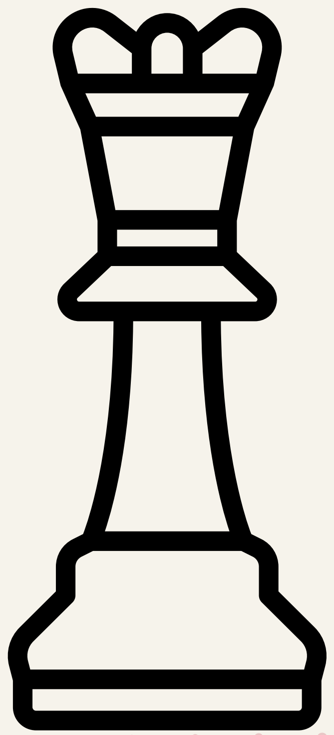


Presented By : SAMIDHA KILLEDAR

ABSTRACT

The N-Queen problem involves placing N queens on an N×N chessboard such that no two queens threaten each other. This project visualizes the solution to the N-Queen problem using the backtracking algorithm in C++. We explore different methods to solve the problem and justify the choice of backtracking due to its efficiency and practicality.

ABOUT THE N-QUEEN PROBLEM



Place N queens on an N×N chessboard so that no two queens can attack each other. A queen can move any number of squares vertically, horizontally, or diagonally. The N-Queen problem is NP-hard because the number of possible configurations increases exponentially with N. There is no known polynomial-time solution for the general case.

SOLVING BY DIFFERENT METHODS

BRUTE FORCE

The brute force approach generates all possible configurations of queens on the board and checks each one to see if it meets the criteria. While this guarantees a solution if one exists, it is extremely slow because it has to check an exponential number of configurations. The time complexity of brute force is also O(N!), making it infeasible for large values of N.

RANDOM PERMUTATIONS

This method of solving involves placing the queens randomly on the board and checking if the configuration is valid. While simple to implement, it is highly inefficient as the probability of randomly finding a valid solution is very low, especially for large N. The time complexity of this approach is O(N!), which is inefficient and impractical for larger boards.

SOLVING BY BACKTRACKING

Backtracking is a more efficient method that incrementally builds solutions and backtracks as soon as it detects that the current solution cannot possibly lead to a valid solution. This method systematically explores the possibilities, placing queens one by one in different rows and columns while ensuring no conflicts. The time complexity of backtracking is significantly lower than random permutations and brute force, making it a practical choice for solving the N-Queen problem.

Why Backtracking is best...?

Backtracking is preferred because it systematically prunes the search space by eliminating invalid configurations early, thereby reducing the number of possibilities that need to be explored. This makes it both efficient and practical for solving the N-Queen problem compared to other methods.

The backtracking algorithm works by placing a queen in the leftmost column and then moving to the next column to place another queen. If a safe position is found, the algorithm moves forward. If no safe position is found, it backtracks to the previous column and tries the next available position. This process continues until all queens are placed or all possibilities are exhausted.

- Pruning: The algorithm eliminates invalid configurations early, reducing the number of possibilities.
- Efficiency: By systematically exploring feasible solutions, the algorithm is much more efficient than random or brute force methods.
- Flexibility: The algorithm can handle different values of N and can be easily adapted for larger board sizes.

DEMONSTRATION

Demonstration: In this section, the code will be shown in action for different values of N. For example, running the solver for N=4 and N=8 will display the solutions and illustrate how the backtracking algorithm efficiently finds the correct placements for the queens. This visual demonstration will highlight the practical application and effectiveness of the algorithm.

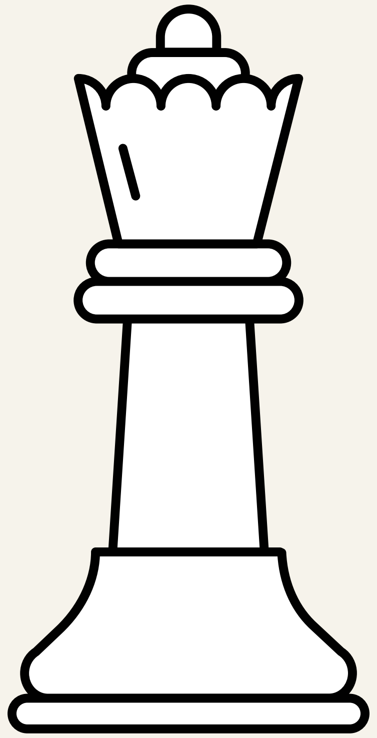
PROBLEMS AND SOLUTIONS

Handling large values of N due to the increased complexity and potential for a large number of invalid configurations.

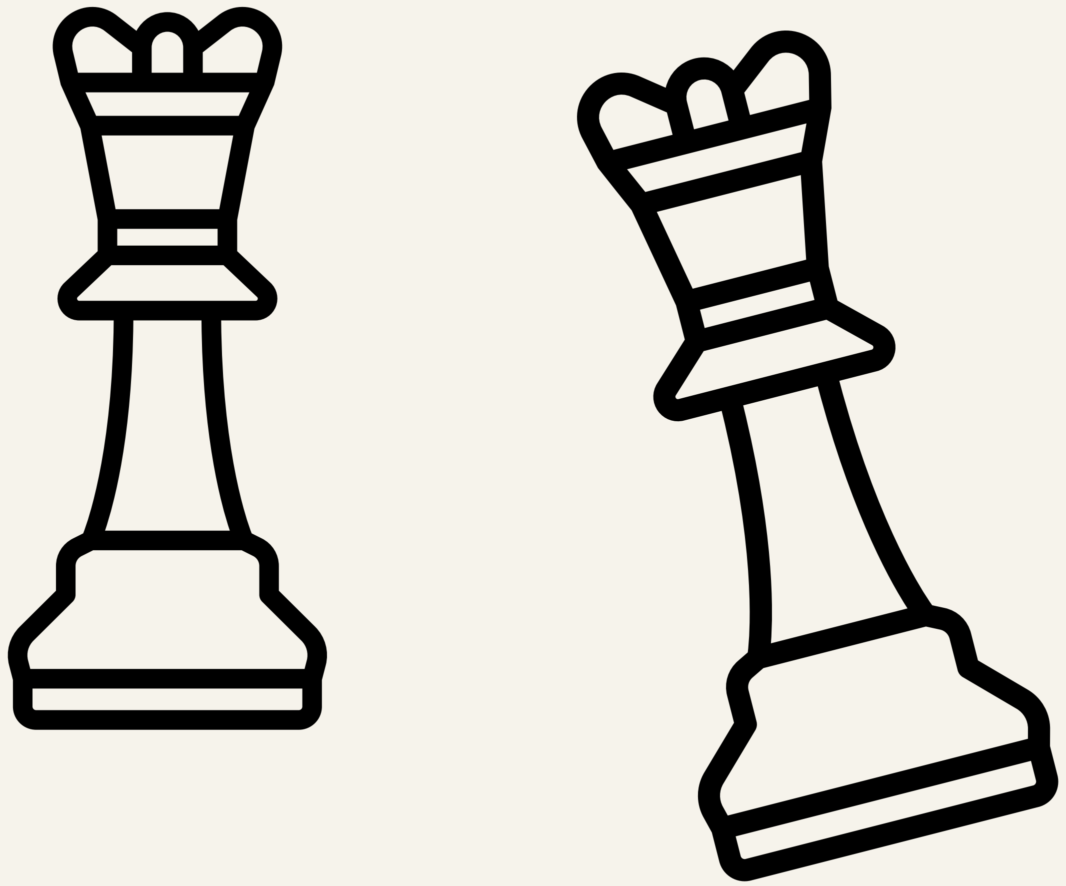
Optimizing the isSafe function to perform faster checks by reducing unnecessary computations.

Ensuring efficient memory usage to handle the board and recursive calls.

Using efficient data structures such as vectors to manage the board state, which helps in reducing the overall memory footprint and improving performance.



N QUEENS



THANK YOU

Presented By : SAMIDHA KILLEDAR