**Chapter 16. Solving the “Eight Queens” Problem by the Recursive Method**

## 16.1 Analysis of the Eight Queens Problem

The eight queens problem on an 8x8 chessboard is a classic example of a puzzle that requires eight queens to be placed so that they do not threaten each other. This means that none of the queens should be on the same file, rank, or diagonal as another queen.

There are several methods to solve this problem, the most common of which is the recursive “Backtracking” algorithm. Sequence of algorithm steps:

1. Determining the starting position for the first queen on the first row and placing it in one of the cells.
2. Move to the next rank and place the second queen, excluding positions threatened by the first.
3. Continuing the process of placing queens, moving to the next rank and eliminating threatening positions.
4. If there are no valid positions left to place the queen on the current rank, you must return to the previous rank and change the position of the last placed queen.
5. Repeat steps 3 and 4 until a solution is found or all possibilities are exhausted.

Thus, the eight queens problem illustrates the use of recursive algorithms and backtracking methods in solving combinatorial problems [18].

## 16.2 Solving the Eight Queens Problem Using the Backtracking Algorithm

The "Backtracking" algorithm ensures that all possible solutions to the eight queens problem on an 8x8 chessboard are found. If only one solution is required, the algorithm can be stopped after it has been found.

The solution process begins by placing the first queen at position A[1][1]. After this, cells that are under threat are marked with crosses. – vertically, horizontally and diagonally (figure 16.1).

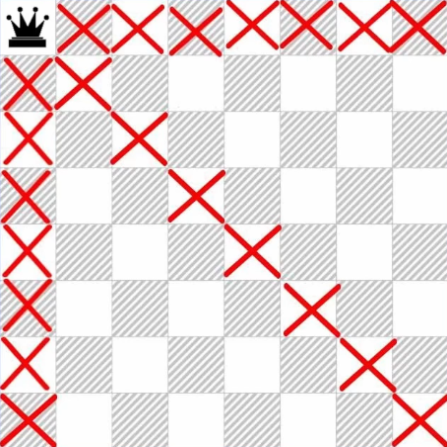


Figure 16.1 – Positions subject to queen attack

The second queen is then placed on the first valid square of the second row, and the threatening positions are also marked (figure 16.2).

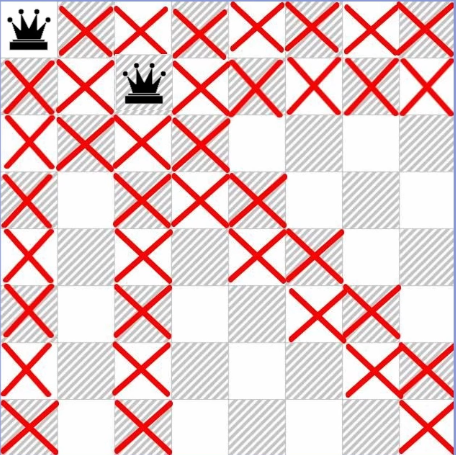


Figure 16.2 – Positions subject to attack by the second queen

This process continues until all eight queens are placed or there are no available squares for the next one (figure 16.3). If, for example, the fifth position cannot accommodate the sixth queen, the algorithm returns to the previous step and tries a different position for the last queen placed. If this does not lead to success, the return to the previous queen continues.

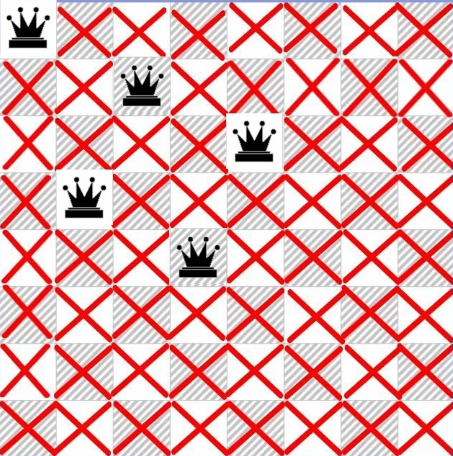


Figure 16.3 – All squares fall under the queen's attack

Thus, the algorithm repeats these steps until all eight queens are placed on the board.

## 16.3 Code for solving the eight queens problem

Section 16.3 reviews the program code in the high-level algorithmic language C++ and proposes a solution to the problem of eight queens on an 8x8 chessboard [6]. Main components of the program are shown in Figure 16.4 and include the constant size\_arr, which specifies the size of the board, and a two-dimensional array arr, which represents the placement of the queens. The attempt variable tracks the number of placement attempts.

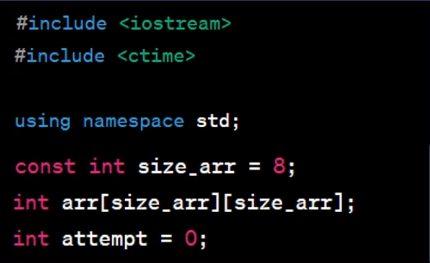


Figure 16.4 – Program code fragment illustrating the main components of the program

The code includes several key functions. The show() function is responsible for displaying the current arrangement of queens, using nested loops to display squares with queens and empty squares (figure 16.5).

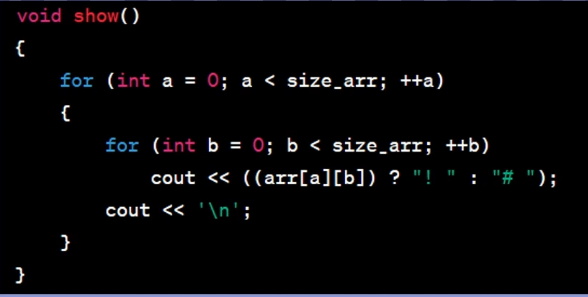


Figure 16.5 – Function for displaying the current coordinates of queens

The check() function checks the possibility of placing a queen at a given position by analyzing the presence of other queens on the verticals, horizontals and diagonals (figure 16.6).

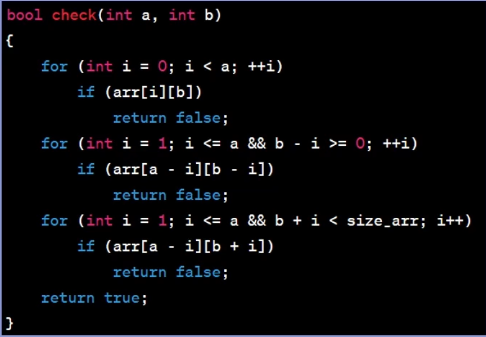


Figure 16.6 – A function that checks whether a queen can be placed

The main logic of the solution is implemented in the recursive function set(), which takes the current horizontal line (line) as an argument. If all queens are successfully placed, the function calls show() to display the solution. Otherwise, it iterates through all possible positions in the current line and checks for queen placement using check(). Once placed successfully, the function is called recursively on the next row, and then the queen is removed from the board to search for other solutions [21].

The main() function starts the queen placement process by calling set(0). At the end of the program, all possible solutions to the eight queens problem are displayed.

Thus, this code effectively finds and displays all solutions to the eight queens problem on an 8x8 chessboard.