```
public class NQueens {

    // Function to print the board
    public static void printBoard(int[][] board) {
        for (int i = 0; i < board.length; i++) {
            for (int j = 0; j < board[i].length; j++) {
                if (board[i][j] == 1) {
                    System.out.print("Q ");
                } else {
                    System.out.print(". ");
                }
            }
            System.out.println();
        }
    }

    // Function to check if a queen can be placed at board[row][col]
    public static boolean isSafe(int[][] board, int row, int col) {
        // Check the column
        for (int i = 0; i < row; i++) {
            if (board[i][col] == 1) {
                return false;
            }
        }

        // Check upper-left diagonal
        for (int i = row - 1, j = col - 1; i >= 0 && j >= 0; i--, j--) {
            if (board[i][j] == 1) {
                return false;
            }
        }

        // Check upper-right diagonal
        for (int i = row - 1, j = col + 1; i >= 0 && j < board.length; i--, j++) {
            if (board[i][j] == 1) {
                return false;
            }
        }

        return true;
    }

    // Function to solve the N Queens problem using backtracking
    public static boolean solveNQueens(int[][] board, int row) {
        // If all queens are placed, return true
        if (row >= board.length) {
            return true;
        }

        // Try all columns in the current row
        for (int col = 0; col < board.length; col++) {
            // Check if it's safe to place queen at board[row][col]
            if (isSafe(board, row, col)) {
                // Place the queen
                board[row][col] = 1;

                // Recursively place queens in the next rows
                if (solveNQueens(board, row + 1)) {
                    return true;
```

```
                }

                // Backtrack if placing queen in the current column doesn't lead to
a solution
                board[row][col] = 0;
            }
        }

        return false; // If no place is found for queen, return false
    }

    public static void main(String[] args) {
        int n = 5; // Number of queens (for 5x5 board)
        int[][] board = new int[n][n]; // Initialize the board with 0

        if (solveNQueens(board, 0)) {
            printBoard(board);
        } else {
            System.out.println("Solution does not exist.");
        }
    }
}
```