```python
def solve_n_queens(n):
    def is_safe(board, row, col):
        # Check this row on left side
        for i in range(col):
            if board[row][i] == 1:
                return False

        # Check upper diagonal on left side
        for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
            if board[i][j] == 1:
                return False

        # Check lower diagonal on left side
        for i, j in zip(range(row, n, 1), range(col, -1, -1)):
            if board[i][j] == 1:
                return False

        return True

    def solve(board, col):
        # Base case: If all queens are placed
        if col >= n:
            solutions.append([row[:] for row in board])
            return True

        res = False
        # Consider this column and try placing this queen in all rows one by one
        for i in range(n):
            if is_safe(board, i, col):
                # Place this queen in board[i][col]
                board[i][col] = 1

                # Make result true if any placement is possible
                res = solve(board, col + 1) or res

                # If placing queen in board[i][col] doesn't lead to a solution,
                # then remove queen from board[i][col]
                board[i][col] = 0  # BACKTRACK

        return res

    # Initialize the board and solutions list
    board = [[0 for _ in range(n)] for _ in range(n)]
    solutions = []
    solve(board, 0)
    return solutions

def print_solutions(solutions):
    for i, solution in enumerate(solutions, 1):
        print(f"Solution {i}:")
        for row in solution:
            print(" ".join("Q" if cell == 1 else "." for cell in row))
        print()

if __name__ == "__main__":
    n = 8  # Standard 8-queens problem
    print(f"Solving {n}-queens problem...\n")
    solutions = solve_n_queens(n)
    print(f"Total solutions found: {len(solutions)}")

    # Uncomment the following line to print all solutions
    print_solutions(solutions)

    # For the 8-queens problem, we expect 92 solutions
    if n == 8:
        assert len(solutions) == 92, "Should find 92 solutions for 8-queens problem"
```