This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

The Complete Solver

```
def solve_mate_in_one(board):
    """
    Find the move that checkmates Black in one move.
    Returns the winning move or None if none exists.
    """
    print("--- Starting Solution Search ---")

    # Generate all possible moves for White
    white_moves = generate_all_moves(board, white_turn=True)

    print(f"Moves to analyze: {len(white_moves)}")

    for move in white_moves:
        r1, c1, r2, c2 = move
        piece = board[r1][c1]

        # Create a copy of the board for simulation
        simulated_board = copy.deepcopy(board)

        # Execute the move in simulation
        simulated_board[r2][c2] = piece
        simulated_board[r1][c1] = '.'

        # Check: does this move checkmate Black?
        if is_checkmate(simulated_board, white_king_in_mate=False):
            print(f"!!! CHECKMATE FOUND !!!")
            print(f"Winning move: {piece} from ({r1},{c1}) to ({r2},{c2})")
            return move

    print("No solution found.")
    return None
```

Co-funded by the European Union

The project "ChessAIThon" (2025-1-ES01-KA220-VET-000354329) is co-funded by the European Union. The views and opinions expressed in this publication are those of the author(s) only and do not necessarily reflect those of the European Union or the Spanish Service for the Internationalisation of Education (SEPIE). Neither the European Union nor the National Agency SEPIE can be held responsible for them.