Lab 3: Chess Bonus

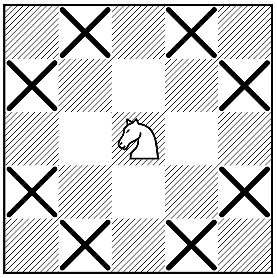
Goal

For bonus points, you will implement various additional features for your chess game from Lab 3. This bonus is divided into several parts which are detailed below. With the exception of the first part, they are all independent and may be completed in any order (or not at all!) Note, however, that **the first part must be completed to be eligible for any bonus points. Your submission must have also successfully passed all the test cases from Parts 1, 2, and 3 of the base Chess Lab to be eligible for any bonus points.**

Part 1: Knight and Queen

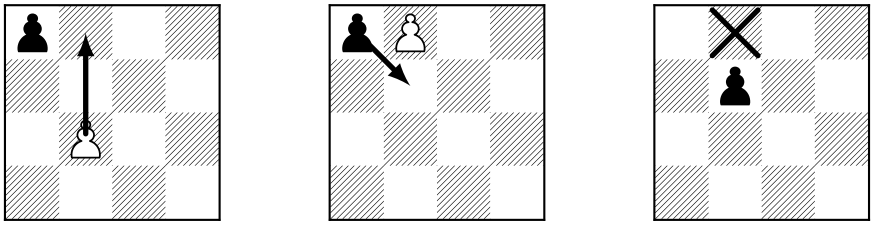
In this part, you will implement two new chess pieces: the Knight (N) and the Queen (Q).

- The queen can move any number of unobstructed spaces horizontally, vertically, or diagonally, and captures in the same way it moves (it essentially acts as the union of a Rook and a Bishop).
- The knight can move two spaces horizontally or vertically, followed by one space in an orthogonal direction. Note that a knight's movement is not obstructed even if there is a piece in the way (i.e. the knight can "jump" over other pieces). In the example below, the knight can move to all the squares marked with a cross:



Part 2: En Passant

French for “in passing”, this chess rule enables a pawn to be captured by another pawn if it moves two spaces and it could have been captured in the space it skipped. Consider the boards below. If the white pawn makes the move indicated, it passes through a square threatened by the black pawn. By the en passant rule, *on the very next turn* the black pawn can move to the space the white pawn moved through and capture it:



Note that if black moved a different piece instead, then on the following turn the en passant rule would not apply, even if the board state still allowed it. Also note that the en passant rule only applies if both the moving piece and capturing piece are pawns.

Part 3: Pawn Promotion

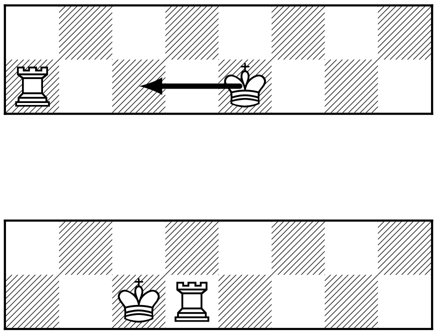
When a pawn reaches the back rank (i.e. row 0 for a white pawn, row $N-1$ for a black pawn) it is “promoted” and replaced by a queen of the same color.

Part 4: Castling

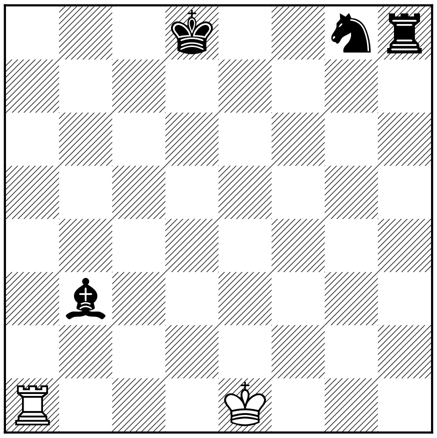
While the king normally moves a single space in any direction, the “castling” rule allows it to move two spaces under circumstances. When castling, the king moves two spaces towards a rook of the same color, and the rook moves to occupy the space the king skips over. Castling is only legal when all of the following conditions have been met:

- The king has not yet moved
- The king is not currently in check
- The rook being castled towards has not yet moved
- The path from the king to the rook is completely unobstructed
- The space the king skips over is not under threat by any piece of an opposing color

In the example below, the king can castle towards the rook on the left, as long as neither piece has yet moved:



In this board, the white king cannot castle as it would pass through a square threatened by the black bishop, and the black king cannot castle as the path to its rook is obstructed by the black knight:



Note that `isValidMove` should return true when attempting to move the king by two spaces if and only if castling in that direction is legal. Additionally, `movePiece` should move both the king and the corresponding roko if the move being made is a legal castle.

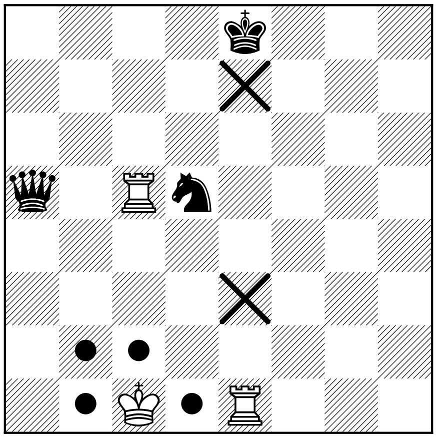
Part 5: Scoring

You will implement a simplified version of the system described here to score board states. To compute the score, first add points for each piece you have on the board:

- 200 points for a King
- 9 points for a Queen
- 5 points for each Rook
- 3 points for each Knight and Bishop
- 1 point for each Pawn

Next, add 0.1 points for every possible legal move you can make. Finally, compute the same score for your opponent and subtract it from your score (i.e. subtract 5 points for each rook of an opposing color, subtract 0.1 points for each legal move your opponent can make, etc.)

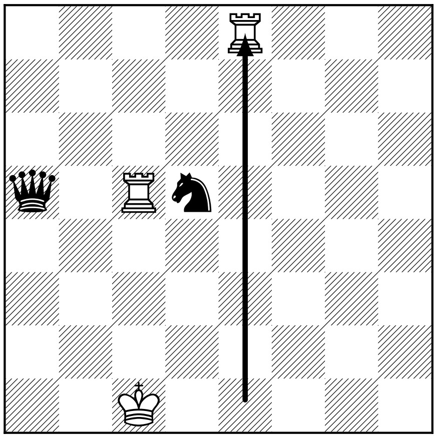
For example, consider the board below:



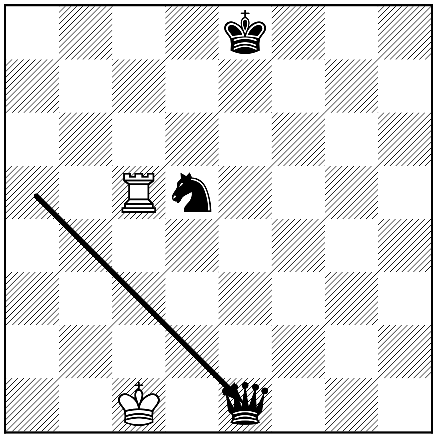
White has a king and two rooks, giving $200 + 5 \times 2 = 210$ points. The king has four legal moves (marked with •s). The top rook has three horizontal and six vertial moves, and the bottom rook has four horizontal and six vertical moves, for a total of $4 + 9 + 10 = 23$ moves. Black has a king, a knight and a queen, giving a total of $200 + 3 + 9 = 212$ points. The queen must move to capture white's rook and the knight has two possible moves it can make to block check (indicated with crosses). The black king can move laterally or diagonally but not down, giving a total of $1 + 2 + 4 = 7$ moves. Thus, white's total score is $210 + 2.3 - 212 - 0.7 = -0.4$ and black's total score is 0.4.

For this part, you must implement two additional functions within the `ChessBoard` class:

- `float scoreBoard()` computes the score from the perspective of the current player (determined by `turn`)
- `float getHighestNextScore()` looks at all possible moves from the perpsective of the current player and returns the highest score the player can achieve after making a single move. For example, white's best move is to capture the black king, resulting in a score of 410.6:



and black's best move is to capture the white rook, resulting in a score of 9.9:



Note that the test cases for part 5 will only use 8 × 8 boards.