Chess Board Problem

Problem: There is a chess board of dimension n X n. You are given with 2 squares on that board S(x1, y1) M(x2, y2). S is a fixed point. M can move diagonally. it can move any number of steps or jumps in 1 move . Find the minimum number of moves M needs to reach S.

Algorithm:

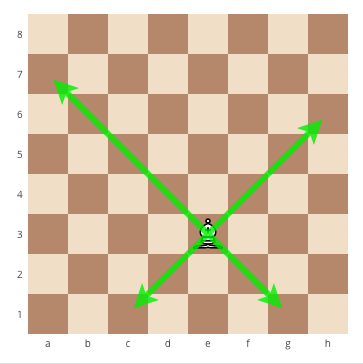
1. Start
2. Start try block
3. **if** (x1 < 0 || x2 < 0 || y1 < 0 || y2 < 0 || x1 > size - 1 ||

x2 > size -1 || y1 > size - 1|| y2 > size - 1) then throw GameException

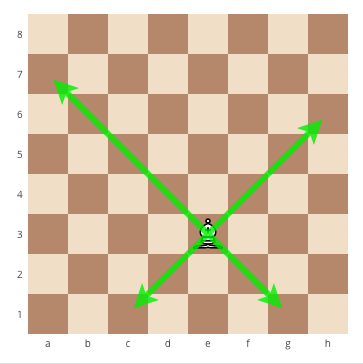
with custom message

1. **if** ((x1 + y1) % 2 != (x2 + y2) % 2) then throw GameException with custom message
2. **if** ((x1 == x2) && (y1 == y2)) then return 0.
3. **if** ( (x1 + y1) == (x2 + y2)) then return 1.
4. **Else return 2**
5. End try
6. Catch block
7. Return maximum integer value
8. Stop

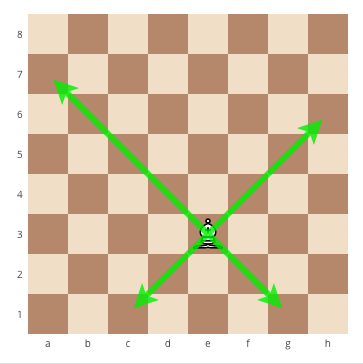
Images: One move required



Two moves required



Collision not possible



Error

