BFS Chess!

Published by Matt in Java ▾

complete algorithms games

You will be given the location of a knight, and an end location. The knight can move in a "L" shape. "L" shape movement means that the knight can change it's `x` coordinate by 2 and it's `y` coordinate by 1 or it can change it's `y` coordinate by 2 and it's `x` coordinate by 1 (you can add and subtract from the x/y).

For example, if the knight is at the position (0, 0), it can move to:

```
(1,2), (1,-2), (2,1), (2,-1), (-1,2), (-1,-2),
```

Your job is to return the least amount of steps needed to go from the position K (knight's start position) to E (end). You will only be given the knight starter coordinates (x1, y1) and the end coordinates (x2, y2).

Constrains: `1 <= x1,y1,x2,y2 <= 8`

Examples

```
knightBFS(1, 1, 8, 8) ➞ 6

knightBFS(1, 1, 3, 2) ➞ 1

knightBFS(8, 8, 3, 3) ➞ 4
```

Notes

ProTip

Ctrl + Enter to check code.

Continue

▲ English

- This is a simplified version of this problem.
- This travel will always be possible.
- The chess board is 8x8.

SUGGEST EDIT