

July 2019 - Challenge

This month's challenge is by Hugo Pfoertner. (Thanks!)

A non-intersecting knight's cycle is a closed tour made of legal moves of a chess knight on a rectangular board of size $m \times n$, such that the straight lines connecting consecutively visited fields make a simple polygon of an enclosed area A .

Find two non-intersecting knight's cycles of length 14 on boards of size ≤ 40 such that the difference between their areas is at least 7.

As an example, there are two non-intersecting knight's cycles of length 4 on boards of size ≤ 16 such that the difference between their areas is 1:

The first has an area of 3 on 4x4 board: a1 c2 d4 b3 a1

and the second has an area of 4 on 3x5 board: c1 e2 c3 a2 c1.

Submit your solution as two lines of 15 visited fields (start=end) using algebraic chess notation as we used above.

Solution:

Both of my length 14 non-intersecting knight's cycles exist on 8x5 boards. The enclosed areas were computed using the shoelace formula and Pick's theorem. https://en.wikipedia.org/wiki/Shoelace_formula

Shoelace formula

Python verifying code for the shoelace formula is included below for reference.

The first cycle with enclosed area 8 is:

a8b6a4c5b3d4c2e1d3e5c4d6b5c7a8

The second cycle with enclosed area 15 is:

a3b5a7c8e7c6e5d3e1c2a1b3d2c4a3

The difference of the two areas is 7 which is at least 7 as required. These cycles are depicted (excepting the last closing path) below:

Drawn with www.cs.umd.edu/class/spring2018/cmsc351/knight_tour/



#The first tour with enclosed area 8 is:
#a8b6a4c5b3d4c2e1d3e5c4d6b5c7a8

```
def PolygonArea(corners):
    n = len(corners) # of corners
    area = 0.0
    for i in range(n):
        j = (i + 1) % n
        area += corners[i][0] * corners[j][1]
        area -= corners[j][0] * corners[i][1]
    area = abs(area) / 2.0
    return area
```

```
corners = [(,5, 7.5), (1.5, 5.5), (,5, 3.5),(2.5, 4.5),(1.5, 2.5),(3.5,3.5),(2.5,1.5),(4.5,,5),(3.5,2.5),(4.5,4.5),(2.5,3.5),(3.5,5.5),
(1.5,4.5),(2.5,6.5) ]
print(PolygonArea(corners))
#Tour area = 8.0
```

#The second tour with enclosed area 15 is:
#A3b5a7c8e7c6e5d3e1c2a1b3d2c4a3

```
def PolygonArea(corners):
    n = len(corners) # of corners
    area = 0.0
    for i in range(n):
        j = (i + 1) % n
        area += corners[i][0] * corners[j][1]
        area -= corners[j][0] * corners[i][1]
    area = abs(area) / 2.0
    return area
```

```
# examples
corners = [(,5, .5), (2.5, 1.5), (4.5, .5),(3.5, 2.5),(4.5, 4.5),(2.5,5.5),(4.5, 6.5),(2.5, 7.5),(,5, 6.5),(1.5, 4.5),(,5,2.5),(2.5, 3.5),(3.5,
1.5),(1.5,2.5) ]
print(PolygonArea(corners))
#Tour area = 15.0
```