

1. PROGRAM 9 INSTRUCTIONS, CS102, PROF. LOFTIN
DUE MONDAY, NOVEMBER 23, 2009

In chess, a knight can move in eight possible directions: From position (i, j) for $0 \leq i, j \leq 7$, the knight can move to one of the positions $(i + 2, j + 1)$, $(i + 2, j - 1)$, $(i + 1, j + 2)$, $(i + 1, j - 2)$, $(i - 1, j + 2)$, $(i - 1, j - 2)$, $(i - 2, j + 1)$, $(i - 2, j - 1)$. So if K represents the knight, the possible positions it can move to are given by an x:

```
. . . . . . . .
. . . . . . . .
. . x . x . . .
. x . . . x . .
. . . K . . . .
. x . . . x . .
. . x . x . . .
. . . . . . . .
```

Write a class `KnightsTour.java` which uses a recursive algorithm to find a *knight's tour* on the chessboard. This means that, starting from the $(0, 0)$ position (the upper left corner), the knight should make moves to visit all the positions of the chessboard exactly once. Here is the output of my program for an 8x8 chessboard:

```
1 12 9 6 3 14 17 20
10 7 2 13 18 21 4 15
31 28 11 8 5 16 19 22
64 25 32 29 36 23 48 45
33 30 27 24 49 46 37 58
26 63 52 35 40 57 44 47
53 34 61 50 55 42 59 38
62 51 54 41 60 39 56 43
```

This tour represents the order in which the squares on the chessboard are visited, 1 through 64.

2. HOW TO DO IT

It is useful to have a global constant representing the size of the board (it is useful in debugging to work with smaller examples). The smallest square board on which there is a knight's tour is 5x5.

```
private final static int BOARD_SIZE = 8;
```

The other global data structures I used in my solution are

```
private int[][] board;
private int num;    // current knight to be placed
```

The `board` array will contain an integer for the correct number of the knight's move (from 1 to `BOARD_SIZE * BOARD_SIZE`), and a zero if the knight has not yet moved there in the current tour.

The `main` method should initialize a new instance of the class `KnightsTour` and begin the recursion at position (0,0), and may contain similar code to the following:

```
KnightsTour K = new KnightsTour();
if (K.tour(0,0))
    System.out.println(K);
else
    System.out.println("No knight's tour possible on a " +
        BOARD_SIZE + " x " + BOARD_SIZE + " board.");
```

Use the constructor to set up the `board` array and initialize `num`. The primary recursive method may look like this:

```
private boolean tour (int row, int column)
```

It should return true if there is a tour going forward placing the current knight at place (row,column). Its format should be similar to the recursive method in the `Maze` class from Lewis & Loftus discussed in lecture. In particular, it will be useful to have a `boolean` variable `done` to signify whether the knight's tour has been found.

Things you should address:

- How to update `num`?
- How to place a knight on the board?
- How to remove a knight from the board (when backtracking)?
- How to check if placing a knight at (row,column) is valid? (It's not valid if you land off the chessboard, and also if this position has already been visited in the current tour.)
- What is the base case of the recursion? How do you check whether the tour is complete?
- How to perform the next move of the knight? The code with the recursive method calls (eight in all, corresponding to the eight possible moves of the knight) should be structured like the recursive calls in `Maze.java`.

You should also write an appropriate `toString()` method.

3. HOW TO TURN IT IN

Turn a stand-alone Java class called `KnightsTour.java` to me by emailing it, as an attachment, to `loftin@rutgers.edu`.

4. DUE DATE

November 23, 2009.

5. HINTS ON DEBUGGING

Set the `BOARD_SIZE` to a small number (4 will be OK for the recursion, though you have to go to at least 5 to get a successful knight's tour). Write in an extra print statement to print out the board whenever it is updated (either by placing a new knight or by removing an old one). This way, you should be able to tell what your code may be doing wrong in a case you can work out by hand.

****VERY IMPORTANT**** Comment out your extra print statements before handing your program in, and set the `BOARD_SIZE` to 8.