

An explanation of Parberry's knight's tour algorithm

This document will explain the algorithm described in Ian Parberry's 1997 paper "An Efficient Algorithm for the Knight's Tour Problem". I will explain the Knight's tour problem, the notation used in the paper, and end with a detailed discussion of the divide and conquer algorithm eluded to in the paper. A reference implementation of the algorithm in Java is provided alongside this document.

My motivation for writing this document stems from the lack of accessible material on Parberry's paper online. I found the paper itself to be very dense and often lacking in explicit notes on points of importance. Other resources available online I also found to be vague or used excessive jargon relating to graph theory. I hope, therefore, to explain the algorithm and the concepts within the paper in a manner suitable for those not familiar with the notation used.

1 The Knight's tour

Parberry describes a tour as "A *knight's tour* is a series of moves made by a knight visiting each square of an $n \times n$ chessboard exactly once". This means we have a knight, who can move in a particular way that we will describe later, and this knight is somewhere on a square board that has some width and some height that we'll call n - our goal in creating a tour is to have this knight move to every square on the board without moving to a square that he's already been to.

Parberry notes that a *knight's tour problem* is about creating a tour for a knight on some specific board size, but the distinction between a knight's tour and a knight's tour *problem* is not important for our purposes and I will use the two interchangeably.

Parberry describes "A knight's tour is called *closed* if the last square visited is also reachable from the first square by a knight's move, and *open* otherwise". For a tour to be closed the starting square must be reachable by a *single* move. We will not include this final move in our discussions or the reference implementation, but it would be valid to do so.

1.1 Tour representation

The representation used in the paper is (verbatim) $G = (V, E)$, where: $V = \{(i, j) \mid 1 \leq i, j \leq n\}$, and $E = \{((i, j), (k, l)) \mid \{|i - k|, |j - l|\} = \{1, 2\}\}$.

This notation is very dense so I'll go through it step by step.

1.1.1 Graphs

G is a graph. A graph is a structure that has vertices and edges. A vertex (the singular of vertices) is a point on the structure that can be accessed from other vertices by moving across an edge. Edges are connections between vertices, and there may or may not be connections between any two vertices. Edges can be (and in this case are) one way - this makes the graph “directed”.

1.1.2 Vertices

In the case of the knight’s tour the vertices represent squares on the board. Recall that Parberry represents the vertices as:

$$V = \{(i, j) \mid 1 \leq i, j \leq n\}$$

This means that every vertex is a pair i and j , where $1 \leq i$ and $j \leq n$.

Parberry does not specify where square (1,1) is located on the board but his later discussion assumes that it is located in the bottom left of the board. If we assume that i is the x axis and j is the y axis then the top left is (1, n). The top right is (n , n), and the bottom right (n , 1).

1.1.3 A refresher on Knights movement

The knight can move 1 square to the left or right, and 2 squares up or down; or 2 squares to the left or right and 1 square up or down. The knight must start at a square and end at a square, this means that he cannot move off the board.

1.1.4 Edges

Edges of a graph (ways of moving between vertices) are the valid moves that the knight can make. Parberry defines each edge as:

$$E = \{((i, j), (k, l)) \mid \{|i - k|, |j - l|\} = \{1, 2\}\}$$

An edge, as Parberry defines it, consists of two points. The first point, on the left, is the square the knight is on (i, j). The second point is the square that the knight ends up on (k, l).

The differences between the start and end squares must be 1 or 2 in the x axis, and the other number of 1 or 2 in the y axis. This means if the knight moves 1 up or down, he must move 2 left or right. Similarly if he moves 2 up or down he must move 1 left or right.

The notation literally reads “The set of the absolute differences between ‘ i and k ’ and ‘ j and l ’ is the set containing 1 and 2”. This is just a concise way of expressing the ways that the knight can move.

1.2 Other relevant tour details

Parberry discusses many other points relating to the Knight’s tour and it’s history in his introduction. He cites many other researchers, some of whose results are duplicated here. Refer to the original paper for more details. Parberry covers *structured* tours and the $n \times (n + 2)$ board tour in the divide and conquer section but I cover it here for cohesive purposes.

- There is a structured closed tour for all boards $n \geq 6$, where n is even.
- There is a structured tour for $n \times (n + 2)$ where $n \geq 6$, and n is even.
- There is an open tour for all boards $n \geq 5$

We do not use the fact that an open tour exists, but it is useful to know if using other methods for creating Knight’s tours.

2 Divide and conquer

2.1 Divide and conquer algorithms generally

Divide and conquer algorithms have:

1. A base case
2. A splitting function
3. A joining function

The base case is the situation in which we stop splitting - these are the cases that we can solve directly. The splitting function is how we divide the problem into smaller problems (with the goal of getting to a base case). The joining function is how we join all of the solved cases back together again.

2.2 The knight's tour

The base case for the knight's tour are a set of structured tours of sizes:

- 6×6
- 6×8
- 8×8
- 8×10
- 10×10
- 10×12
- 6×7
- 7×8
- 8×9
- 9×10
- 10×11
- 11×12

The idea here is to gradually reduce the problem, through dividing the board, to one that we can solve. In this case, we can solve a large number of problems - and because they are all structured solutions we can put them back together again.

Parberry provides his algorithm in the form of a proof. I will restate his proof as steps in an algorithm, and then explain why each step is needed.

Parberry defines k to be $\frac{n}{4}$. Where n is divisible by 4, the four quadrants are $2k \times 2k$ squares. That is, we divide the board in half in each direction.

Each side divisible by $\frac{n-2}{4}$ is divided into $2k \times 2(k+1)$ or $2(k+1) \times 2k$. We determine which to use by which fits within the dimensions. The layout is then, for $A = 2k$, $B = 2k + 2$ where $k = \frac{n-2}{4}$ A x B, B x B, A x A, and B x A going from the top left to the bottom right.

Each side divisible by $n = 4k + 2$ is divided into the same as previously but in a different arrangement: A x A, A x B, A x A, A x B.

The degenerate case where n is divisible by 4 is where $A = B$ and either layout may be used.