

Knight's tour

From Wikipedia, the free encyclopedia

The **knight's tour** is a mathematical problem involving a knight on a chessboard. The knight is placed on the empty board and, moving according to the rules of chess, must visit each square exactly once. A knight's tour is called a *closed tour* if the knight ends on a square attacking the square from which it began (so that it may tour the board again immediately with the same path). Otherwise the tour is *open*. The exact number of open tours is still unknown. Creating a program to solve the knight's tour is a common problem given to computer science students.^[1] Variations of the knight's tour problem involve chessboards of different sizes than the usual 8×8 , as well as irregular (non-rectangular) boards.

Contents

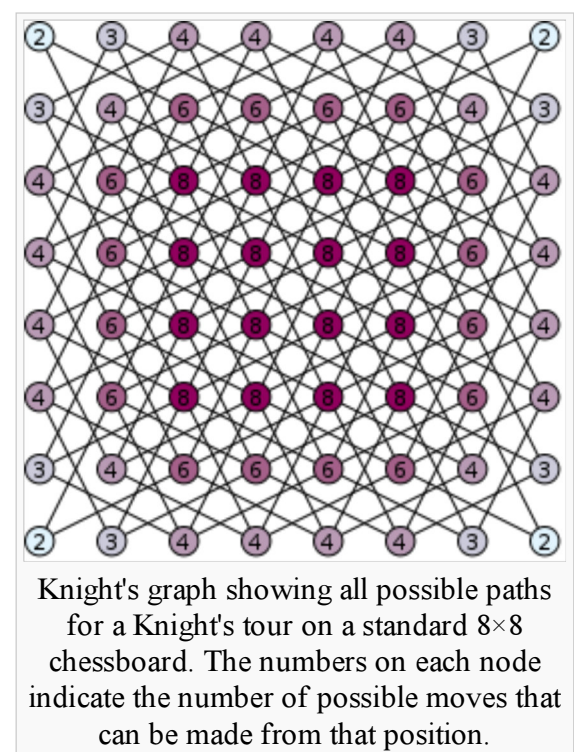
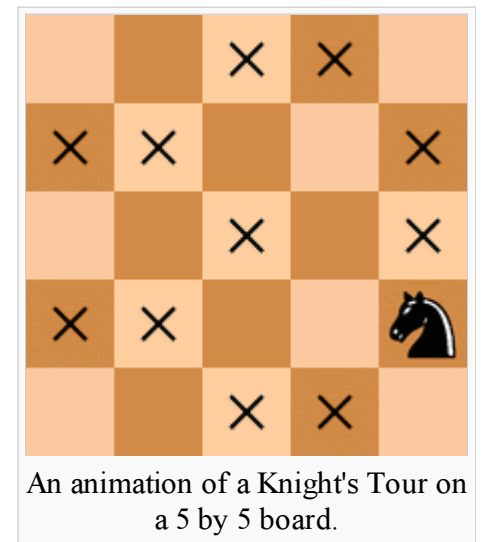
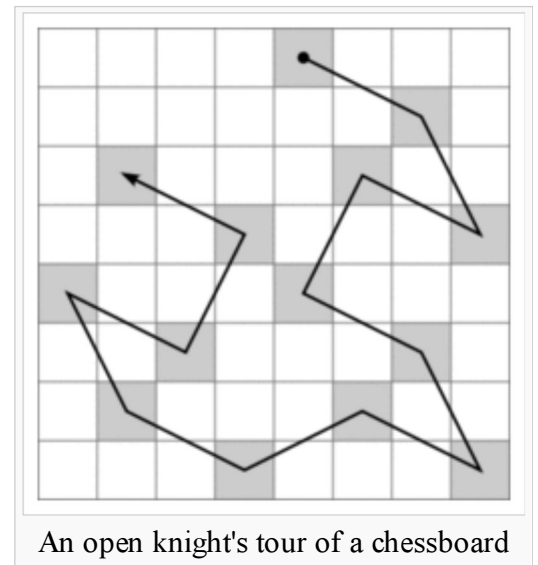
- 1 Theory
- 2 History
- 3 Closed tours
 - 3.1 Schwenk's Theorem
 - 3.1.1 Condition 1
 - 3.1.2 Condition 2
 - 3.1.3 Condition 3
 - 3.1.4 All other cases
- 4 Computer algorithms
 - 4.1 Neural network solutions
 - 4.2 Warnsdorff's algorithm
 - 4.2.1 Algorithm
- 5 See also
- 6 Notes
- 7 External links
 - 7.1 Implementations

Theory

The knight's tour problem is an instance of the more general Hamiltonian path problem in graph theory. The problem of finding a closed knight's tour is similarly an instance of the hamiltonian cycle problem. Note however that, unlike the general Hamiltonian path problem, the knight's tour problem can be solved in linear time.^[2]

History

The earliest known references to the Knight's Tour problem date back to the 9th century CE. The pattern of a knight's tour on a half-board has been presented in verse form (as a literary constraint) in the highly stylized Sanskrit poem *Kavyalankara*^[3] written by the 9th century Indian poet Rudrata, which discusses the art of poetry, especially with relation to theater (Natyashastra). As was often the practice in ornate Sanskrit poetry, the syllabic patterns of this poem



elucidate a completely different motif, in this case an open knight's tour on a half-chessboard.

One of the first mathematicians to investigate the knight's tour was Leonhard Euler. The first algorithm for completing the Knight's Tour was Warnsdorff's algorithm, first described in 1823 by H. C. Warnsdorff.

In the 20th century the Oulipo group of writers used it among many others. The most notable example is the 10×10 Knight's Tour which sets the order of the chapters in Georges Perec's novel *Life: A User's Manual*. The sixth game of the 2010 World Chess Championship between Viswanathan Anand and Veselin Topalov saw Anand making 13 consecutive knight moves (albeit using both knights) -- online commentators jested that Anand was trying to solve the Knight's Tour problem during the game.

Closed tours

On an 8×8 board, there are exactly 26,534,728,821,064 directed closed tours (i.e. two tours along the same path that travel in opposite directions are counted separately).^{[4][5][6]} The number of *undirected* closed tours is half this number, since every tour can be traced in reverse. There are 9,862 undirected closed tours on a 6×6 board.^[7] No closed tours exist for smaller square boards (this is a corollary of the following theorem).

Schwenk's Theorem

For any $m \times n$ board with m less than or equal to n , a **closed** knight's tour is always possible **unless** one or more of these three conditions are true:

1. m and n are both odd; m and n are not both 1
2. $m = 1, 2$, or 4 ; m and n are not both 1
3. $m = 3$ and $n = 4, 6$, or 8 .

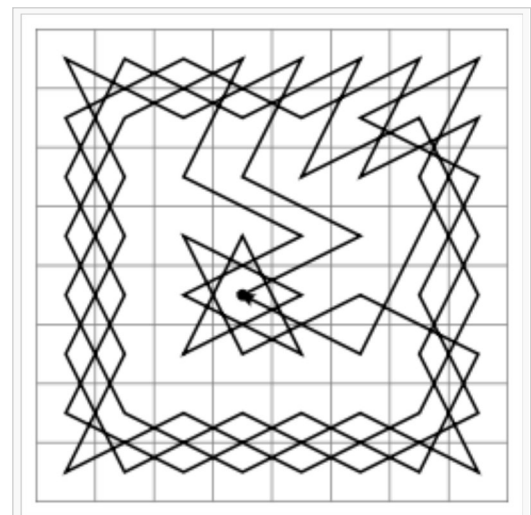
Condition 1

One can show that condition 1 prohibits closed tours by a simple argument based on parity and coloring. For the standard black-and-white coloring of the chessboard, the knight must move either from a black square to a white square or from a white square to a black square. Thus in a closed tour the knight must visit the same number of white squares as black squares, and the number of squares visited in total must therefore be even. But if m and n are both odd, the total number of squares is odd. (For example, in a 5×5 checkerboard there are 13 of one color and 12 of the other.) Therefore closed tours do not exist. Open tours may still exist (with the exception of the trivial case 1×1).

Condition 2

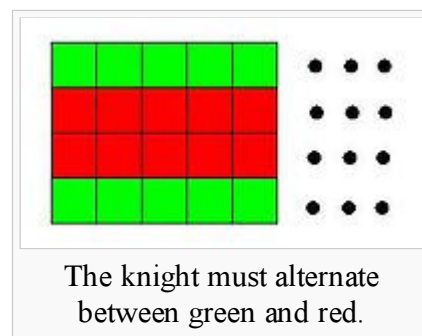
Condition 2 states that when the shorter side is of length 1, 2, or 4, no closed tour is possible.

When $m = 1$ or 2 , no closed tour is possible because the knight would not be able to reach every square (once again, with the exception of the trivial case 1×1). It can be shown that a $4 \times n$ board cannot have a closed tour either, by using a coloring argument.



The Knight's tour as solved by The Turk, a chess-playing machine hoax. This particular solution is closed (circular), and can be completed from any point on the board.

Start by assuming that a $4 \times n$ board has a closed knight's tour. Let A_1 be the set of all squares that would be black and A_2 all the squares that would be white, if they were colored according to the alternating black-and-white checkerboard scheme. Consider the figure at right. Define B_1 to be the set of green squares and B_2 as the set of red squares. There are an equal number of red squares as green squares. Note that from a square in B_1 the knight must next jump to a square in B_2 . And since the knight must visit every square once, when the knight is on a square in B_2 it must move to a square in B_1 next (otherwise the knight will need to travel to two squares in B_1 later).



If we follow the hypothetical knight's tour we will find a contradiction.

1. The first square the knight goes to will be a square of A_1 and B_1 . If it is not, we switch the definitions of B_1 and B_2 so that it is true.
2. The second square must be an element of the sets A_2 and B_2 .
3. The third square must be an element of A_1 and B_1 .
4. The fourth square must be an element of the sets A_2 and B_2 .
5. And so on.

It follows that set A_1 has the same elements as set B_1 , and set A_2 has the same elements as set B_2 . But this is obviously not true, as the red and green pattern shown above is not the same as a checkerboard pattern; the set of red squares is not the same as the set of black squares (or white, for that matter).

So our above assumption was false and there are no closed knight's tours for any $4 \times n$ board, for any n .

Condition 3

Condition 3 may be proved using casework. Attempting to construct a closed knight's tour on a 3 by 4, 3 by 6, or 3 by 8 will lead to definite failure. 3 by n boards with n even and greater than 8 can be shown to have closed tours by induction (a repeating pattern).

All other cases

Simply proving the above three conditions does not prove the theorem; it is still required to prove that all rectangular boards that *do not* fall in one of the above three categories have closed knight's tours.^[8]

Computer algorithms

Algorithms other than the simple brute-force search to find knight's tour solutions are discussed below. It is important to note that exhaustive brute force approach (one which iterates through all possible 64-long move sequences) can *never* be applied to the Knight's Tour problem. There are approximately 4×10^{51} such sequences,^[9] and it would take an unfathomable amount of time to simulate through such a large number of sequences.

Neural network solutions

The Knight's Tour problem also lends itself to being solved by a neural network implementation.^[10] The network is set up such that every legal knight's move is represented by a neuron, and each neuron is initialized randomly to be either "active" or "inactive" (output of 1 or 0), with 1 implying that the neuron is part of the final solution. Each neuron also has a state function (described below) which is initialized to 0.

When the network is allowed to run, each neuron can change its state and output based on the states and outputs of its neighbors (adjacent knight's moves) according to the following transition rules:

$$U_{t+1}(N_{i,j}) = U_t(N_{i,j}) + 2 - \sum_{N \in G(N_{i,j})} V_t(N)$$

$$V_{t+1}(N_{i,j}) = \begin{cases} 1 & \text{if } U_{t+1}(N_{i,j}) > 3 \\ 0 & \text{if } U_{t+1}(N_{i,j}) < 0 \\ V_t(N_{i,j}) & \text{otherwise,} \end{cases}$$

where t represents discrete intervals of time, $U(N_{i,j})$ is the state of the neuron connecting square i to square j , $V(N_{i,j})$ is the output of the neuron from i to j , and $G(N_{i,j})$ is the set of neighbors of the neuron.

Although divergent cases are possible, the network should eventually converge, which occurs when no neuron changes its state from time t to $t + 1$. When the network converges, a solution is found. The solution found by the network will be either a knight's tour, or a series of two or more independent knight's tours within the same board.

Warnsdorff's algorithm

Warnsdorff's algorithm is a heuristic method for solving the knight's tour, based on the rule of choosing the square among those immediately accessible by the knight move that would give the fewest possible knight's moves following the move to that square. In graph-theoretic terms, each move is made to the adjacent vertex with the least degree. (Pohl has devised a rule for breaking ties.) This algorithm may also more generally be applied to any graph. Although the Hamiltonian path problem is NP-hard in general, on many graphs that occur in practice this heuristic is able to successfully locate a solution in linear time.^[11] The knight's tour is a special case.^[12]

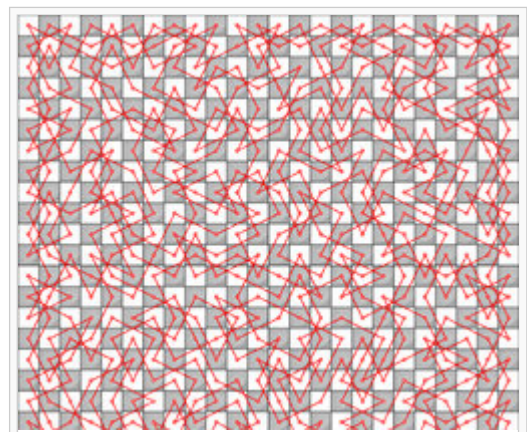
The algorithm was first described in "Des Rösselsprungs einfachste und allgemeinste Lösung" by H. C. Warnsdorff in 1823.^[12]

Algorithm

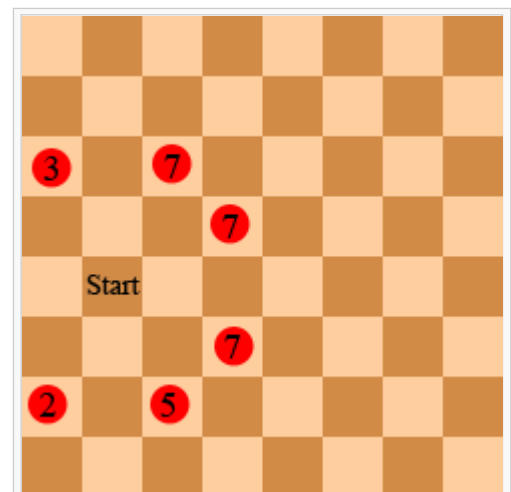
Warnsdorff's algorithm will do for any initial position of the knight on the board. All the possible squares which may be reached in one move from this position are located, and the number of moves that the knight would be able to make from each of these squares is found.^[13] Then, the algorithm dictates that the knight move to the square that has the least number of possible following moves. The process is then repeated until each square has been visited.^{[12][14]}

Some definitions:

- A position Q is accessible from a position P if P can move to Q by a single knight's move, and Q has not yet been visited.



Closed Knight's Tour on a 24×24 board solved by a neural network.



A graphical representation of the Knight's Tour implemented using Warnsdorff's Algorithm. The numbers in red circles denote the number of next possible moves that the knight could make if it moves from the start position to the one with the circle on it.

- The accessibility of a position P is the number of positions accessible from P.

Algorithm:

1. set P to be a random initial position on the board
2. mark the board at P with the move number "1"
3. for each move number from 2 to the number of squares on the board:
 1. let S be the set of positions accessible from the input position
 2. set P to be the position in S with minimum accessibility
 3. mark the board at P with the current move number
4. return the marked board – each square will be marked with the move number on which it is visited.

See also

- Abu-Bakr Muhammad ben Yahya as-Suli
- George Koltanowski
- Longest uncrossed knight's path

Notes

1. ^ H. M. Deitel, P. J. Deitel. "Java How To Program Fifth Edition." *Prentice Hall*, Upper Saddle River, New Jersey, pp. 326-328. 2003.
2. ^ A. Conrad, T. Hindrichs, H. Morsy, and I. Wegener. "Solution of the Knight's Hamiltonian Path Problem on Chessboards." *Discrete Applied Math*, volume 50, no.2, pp.125-134. 1994.
3. ^ Satyadev, Chaudhary. *Kavyalankara of Rudrata (Sanskrit Text, with Hindi translation)*; . Delhi: Parimal Sanskrit Series No. 30.
4. ^ Martin Loebbing; Ingo Wegener (1996). "The Number of Knight's Tours Equals 33,439,123,484,294 — Counting with Binary Decision Diagrams" (http://www.combinatorics.org/Volume_3/Abstracts/v3i1r5.html) . *The Electronic Journal of Combinatorics* **3** (1): R5. http://www.combinatorics.org/Volume_3/Abstracts/v3i1r5.html. **Remark:** The authors later admitted (http://www.combinatorics.org/Volume_3/Comments/v3i1r5.html) that the announced number is incorrect. According to McKay's report, the correct number is 13,267,364,410,532 and this number is repeated in Wegener's 2000 book.
5. ^ Brendan McKay (1997). "Knight's Tours on an 8x8 Chessboard" (http://www.combinatorics.org/Volume_3/Comments/v3i1r5.01.ps) . *Technical Report TR-CS-97-03* (Department of Computer Science, Australian National University). http://www.combinatorics.org/Volume_3/Comments/v3i1r5.01.ps.
6. ^ Wegener, I. (2000). *Branching Programs and Binary Decision Diagrams* (<http://books.google.com/?id=DZjVz9E4f8C&pg=PA369&dq=532>) . Society for Industrial & Applied Mathematics. ISBN 0-898-71458-3. <http://books.google.com/?id=DZjVz9E4f8C&pg=PA369&dq=532>.
7. ^ Weisstein, Eric W., "Knight's Tour (<http://mathworld.wolfram.com/KnightsTour.html>) " from MathWorld.
8. ^ Watkins, John J. (2004). *Across the Board: the Mathematics of Chessboard Problems*. Princeton University Press. ISBN 0691115036.
9. ^ "Enumerating the Knight's Tour" (<http://www.josiahland.com/?p=781>) . <http://www.josiahland.com/?p=781>.
10. ^ Y. Takefuji, K. C. Lee. "Neural network computing for knight's tour problems." *Neurocomputing*, 4(5):249-254, 1992.
11. ^ Pohl, Ira (July 1967). "A method for finding Hamilton paths and Knight's tours" (<http://portal.acm.org/citation.cfm?id=363463>) . *Communications of the ACM* **10** (7): 446–449. doi:10.1145/363427.363463 (<http://dx.doi.org/10.1145%2F363427.363463>) . <http://portal.acm.org/citation.cfm?id=363463>.
12. ^ ^a ^b ^c Alwan, Karla; Waters, K. (1992). "Finding Re-entrant Knight's Tours on N-by-M Boards" (<http://portal.acm.org/citation.cfm?id=503806>) (PDF). ACM Southeast Regional Conference. New York, New York: ACM. pp. 377–382. doi:10.1145/503720.503806 (<http://dx.doi.org/10.1145%2F503720.503806>) . <http://portal.acm.org/citation.cfm?id=503806>. Retrieved 2008-10-28.
13. ^ Tömberg, Gunno (1999). "Warnsdorff's rule" (<http://web.telial.com/~u85905224/knight/eWarnsd.htm>) . <http://web.telial.com/~u85905224/knight/eWarnsd.htm>. Retrieved 2008-09-25.
14. ^ Cancela, Hector; Mordecki, Ernesto (2006-08-31) (PDF). *Counting Knight's Tours through the Randomized*

Warnsdorff Rule (<http://www.cmat.edu.uy/~mordecki/articles/warnsdorff.pdf>) . <http://www.cmat.edu.uy/~mordecki>

</articles/warnsdorff.pdf>. Retrieved 2008-10-18.

External links

- Richard W. Henderson, Eugene Roger Apodaca A Knight of Egodeth: Zen Raptured Quietude (<http://www.amazon.com/dp/0979763002>) , 240 Solutions to the Knights Tour in form of Game Book
- Warnsdorff's Rule (<http://web.telia.com/~u85905224/knight/eWarnsd.htm>) and its efficiency (<http://web.telia.com/~u85905224/knight/bWarnsd.htm>) from Warnsdorff's Rule Web Page
- Mario Velucchi The ultimate Knight's Tour page of Links (<http://www.velucchi.it/mathchess/knight.htm>)
- The knight's tour (<http://www.borderschess.org/KnightTour.htm>)
- Knight's tour notes (<http://www.ktn.freeuk.com/sitemap.htm>)
- Sloane's Integer Sequence A001230 (<http://oeis.org/A001230>)
- Preprint of Pohl's paper (freely accessible) (<http://www.slac.stanford.edu/cgi-wrap/getdoc/slac-pub-0261.pdf>)
- SilverKnight (<http://www.papnkukn.net/silverknight/>) - online knight's tour game
- Knightfall (<http://riddler.batbad.com/riddles/knightfall/index.html>) - online version of the game as presented by the Riddler.

Implementations

- The Knight's Tour (<http://demonstrations.wolfram.com/TheKnightsTour/>) by Jay Warendorff, Wolfram Demonstrations Project
- A Simple backtracking implementation in C++ (<http://www.compgeom.com/~piyush/teach/3330/homeworks/knightour.cpp>)
- A Simple implementation in standard Prolog (<http://junk.kymhorskell.com/knight.html>)
- An implementation in C# (<http://www.knightstour.co.uk>)
- Knight's Tours Using a Neural Network (<http://dmitrybrant.com/knights-tour>) Program that creates tours using a neural network, plus gallery of images.

Retrieved from "http://en.wikipedia.org/wiki/Knight%27s_tour"

Categories: Graph algorithms | Chess and mathematics | Chess problems | Graph theory | Mathematical problems

- This page was last modified on 27 December 2010 at 03:54.
 - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
- Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.