Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

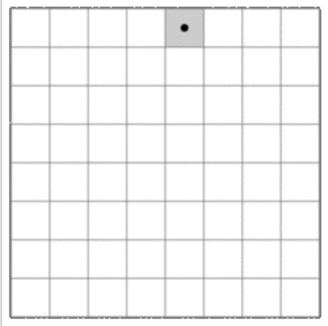
Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
العربية
Català
Čeština
Deutsch
Español
★ فارسی
Français
Frysk
한국어
हिन्दी
Bahasa Indonesia
Italiano
עברית
Latina
■■■■■■
Nederlands
日本語
Polski
Português
Русский
Slovenščina
Ślůnski
Српски / srpski
Suomi
ไทย
Türkçe
Українська
Tiếng Việt
中文
Edit links

Knight's tour

From Wikipedia, the free encyclopedia
(Redirected from Knight's Tour)



An open knight's tour of a chessboard

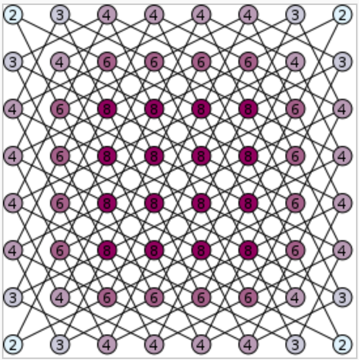
An animation of a knight's tour on a 5 by 5 board.

A **knight's tour** is a sequence of moves of a knight on a chessboard such that the knight visits every square only once. If the knight ends on a square that is one knight's move from the beginning square (so that it could tour the board again immediately, following the same path), the tour is *closed*, otherwise it is *open*.

The **knight's tour problem** is the mathematical problem of finding a knight's tour. Creating a program to find a 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

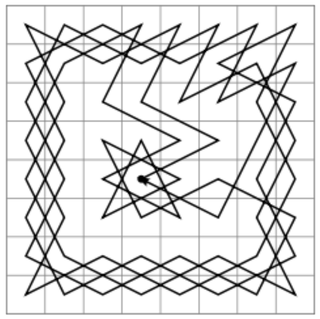
Theory [edit]



Knight's graph showing all possible paths for a knight's tour on a standard 8×8 chessboard. The numbers on each node indicate the number of possible moves that can be made from that position.

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. Unlike the general Hamiltonian path problem, the knight's tour problem can be solved in linear time.[2]

History [edit]



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

The earliest known reference to the knight's tour problem dates back to the 9th century AD. In Rudraṭa's *Kavyalankara*[3] (5.15), a Sanskrit work on Poetics, the pattern of a knight's tour on a half-board has been presented as an elaborate poetic figure ("citra-alaṅkāra") called the "turagapadabandha" or 'arrangement in the steps of a horse.' The same verse in four lines of eight syllables each can be read from left to right or by following the path of the knight on tour. Since the Indic writing systems used for Sanskrit are syllabic, each syllable can be thought of as representing a square on a chess board. Rudrata's example is as follows:

से ना ली ली ली ना ना ना ली

ली ना ना ना ना ली ली ली ली

न ली ना ली ली ले ना ली ना

ली ली ली ना ना ना ना ना ली

se nā lī lī lī nā nā lī

lī nā nā nā nā lī lī lī

na lī nā lī le nā lī nā

lī lī lī nā nā nā nā lī

For example, the first line can be read from left to right or by moving from the first square to second line, third syllable (2.3) and then to 1.5 to

2.7 to 4.8 to 3.6 to 4.4 to 3.2.

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

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 commentors jested that Anand was trying to solve the Knight's Tour problem during the game.

Existence [edit]

Schwenk[4] proved that for any $m \times n$ board with $m \leq n$, a closed knight's tour is always possible *unless* one or more of these three conditions are met:

1. m and n are both odd
2. m = 1, 2, or 4
3. m = 3 and n = 4, 6, or 8.

Cull *et al.* and Conrad *et al.* proved that on any rectangular board whose smaller dimension is at least 5, there is a (possibly open) knight's tour.[5][2]

Number of tours [edit]

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, as are rotations and reflections).[6][7][8] 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.[9]

The number of directed open tours on an $n \times n$ board for n = 1, 2, … are:

1, 0, 0, 0, 1728, 6637920, 165575218320, 19591828170979904. (sequence A165134 in OEIS).

Finding tours with computers [edit]

There are quite a number of ways to find a knight's tour on a given board with a computer. Some of these methods are algorithms while others are heuristics.

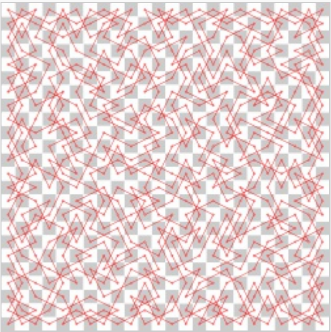
Brute force algorithms [edit]

A brute-force search for a knight's tour is impractical on all but the smallest boards;[10] for example, on an 8x8 board there are approximately 4×10^{51} possible move sequences,[11] and it is well beyond the capacity of modern computers (or networks of computers) to perform operations on such a large set. However the size of this number gives a misleading impression of the difficulty of the problem, which can be solved "by using human insight and ingenuity ... without much difficulty."[10]

Divide and conquer algorithms [edit]

By dividing the board into smaller pieces, constructing tours on each piece, and patching the pieces together, one can construct tours on most rectangular boards in polynomial time.[5][12]

Neural network solutions [edit]



Closed knight's tour on a 24 × 24 board solved by a neural network.

The Knight's Tour problem also lends itself to being solved by a neural network implementation.[13] 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 (those exactly one knight's move away) according to the following transition rules:

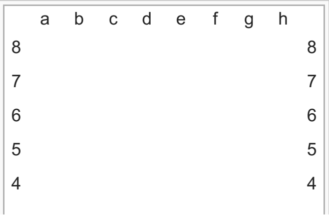
$$U_{t+1}(N_{i,j}) = U_t(N_{i,j}) + 2 - \sum_{N \in G(N_{i,j})} V_t(N)$$

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

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, either the network encodes a knight's tour or a series of two or more independent circuits within the same board.

Warnsdorf's rule [edit]

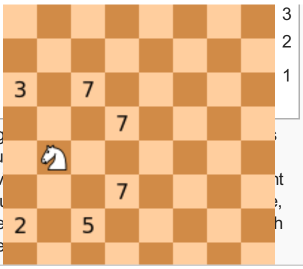


Warnsdorf's rule is a heuristic for finding a knight's tour. We move the knight so that we always proceed to the square from which the knight will have the *fewest* onward moves. When calculating the number of onward moves for each candidate square, we do not count moves that revisit any square already visited. It is, of course, possible to have two or more choices for which the number of onward moves is equal; there are various methods for breaking such ties, including one devised by Pohl [14] and another by Squirrel and Cull.[15]

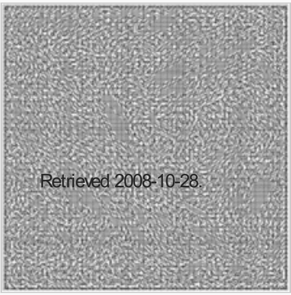
This rule may also more generally be applied to any graph. In graph-theoretic terms, each move is made to the adjacent vertex with the least degree. Although the Hamiltonian path problem is NP-hard in general, on many graphs that occur in practice this heuristic is able to successfully locate a solution in linear time.[14] The knight's tour is a special case.[16]

The heuristic was first described in "Des Rösselsprungs einfachste und allgemeinste Lösung" by H. C. von Warnsdorf in 1823.[16] A computer program that finds a Knight's Tour for any starting position using Warnsdorf's rule can be found in the book 'Century/Acorn User Book of Computer Puzzles' edited by Simon Dally (ISBN 071260541X).

3 3
2 2
1 1



Ag ;
Ru
giv nt
cou e,
the h
the



A very large (130x130) square open Knight's Tour created using Warnsdorff's Rule.

See also [edit]

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

Notes [edit]

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* *b* Conrad, A.; Hindrichs, T.; Morsy, H. & Wegener, I. (1994). "Solution of the Knight's Hamiltonian Path Problem on Chessboards". *Discrete Applied Mathematics* **50** (2): 125–134. doi:10.1016/0166-218X(92)00170-Q.
3. ^ Satyadev, Chaudhary. *Kavyalankara of Rudrata (Sanskrit Text, with Hindi translation);*. Delhitraversal: Parimal Sanskrit Series No. 30.
4. ^ Allen J. Schwenk (1991). "Which Rectangular Chessboards Have a Knight's Tour?". *Mathematics Magazine*: 325–332.
5. ^ *a* *b* Cull, P.; De Curtins, J. (1978). "Knight's Tour Revisited" (PDF). *Fibonacci Quarterly* **16**: 276–285.
6. ^ Martin Loebbing; Ingo Wegener (1996). "The Number of Knight's Tours Equals 33,439,123,484,294 — Counting with Binary Decision Diagrams". *The Electronic Journal of Combinatorics* **3** (1): R5. **Remark:** The authors later admitted 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.
7. ^ Brendan McKay (1997). "Knight's Tours on an 8x8 Chessboard". *Technical Report TR-CS-97-03* (Department of Computer Science, Australian National University).
8. ^ Wegener, I. (2000). *Branching Programs and Binary Decision Diagrams*. Society for Industrial & Applied Mathematics. ISBN 0-89871-458-3.
9. ^ Weisstein, Eric W., "Knight's Tour", *MathWorld*.
10. ^ *a* *b* Simon, Dan (2013), *Evolutionary Optimization Algorithms*, John Wiley & Sons, pp. 449–450, ISBN 9781118659502, "The knight's tour problem is a classic combinatorial optimization problem. ... The cardinality N_x of x (the size of the search space) is over 3.3×10^{13} (Löbbing and Wegener, 1995). We would not want to try to solve this problem using brute force, but by using human insight and ingenuity we can solve the knight's tour without much difficulty. We see that the cardinality of a combinatorial optimization problem is not necessarily indicative of its difficulty."
11. ^ "Enumerating the Knight's Tour".
12. ^ Parberry, Ian (1997). "An Efficient Algorithm for the Knight's Tour Problem" (PDF). *Discrete Applied Mathematics* **73**: 251–260. doi:10.1016/S0166-218X(96)00010-8.
13. ^ Y. Takefuji, K. C. Lee. "Neural network computing for knight's tour problems." *Neurocomputing*, 4(5):249–254, 1992.
14. ^ *a* *b* Pohl, Ira (July 1967). "A method for finding Hamilton paths and Knight's tours". *Communications of the ACM* **10** (7): 446–449. doi:10.1145/363427.363463.
15. ^ Squirrel, Douglas; Cull, P. (1996). "A Warnsdorff-Rule Algorithm for Knight's Tours on Square Boards" (PDF). Retrieved 2011-08-21.
16. ^ *a* *b* Alwan, Karla; Waters, K. (1992). *Finding Re-entrant Knight's Tours on N-by-M Boards* (PDF). ACM Southeast Regional Conference. New York, New York: ACM. pp. 377–382. doi:10.1145/503720.503806.

External links [edit]

- (sequence A001230 in OEIS)

H. C. von Warnsdorf 1823 in Google Books

Wikimedia Commons has media related to ***Knight's Tours***.

Categories: Graph algorithms | Mathematical chess problems | Chess problems | Hamiltonian paths and cycles | Mathematical problems