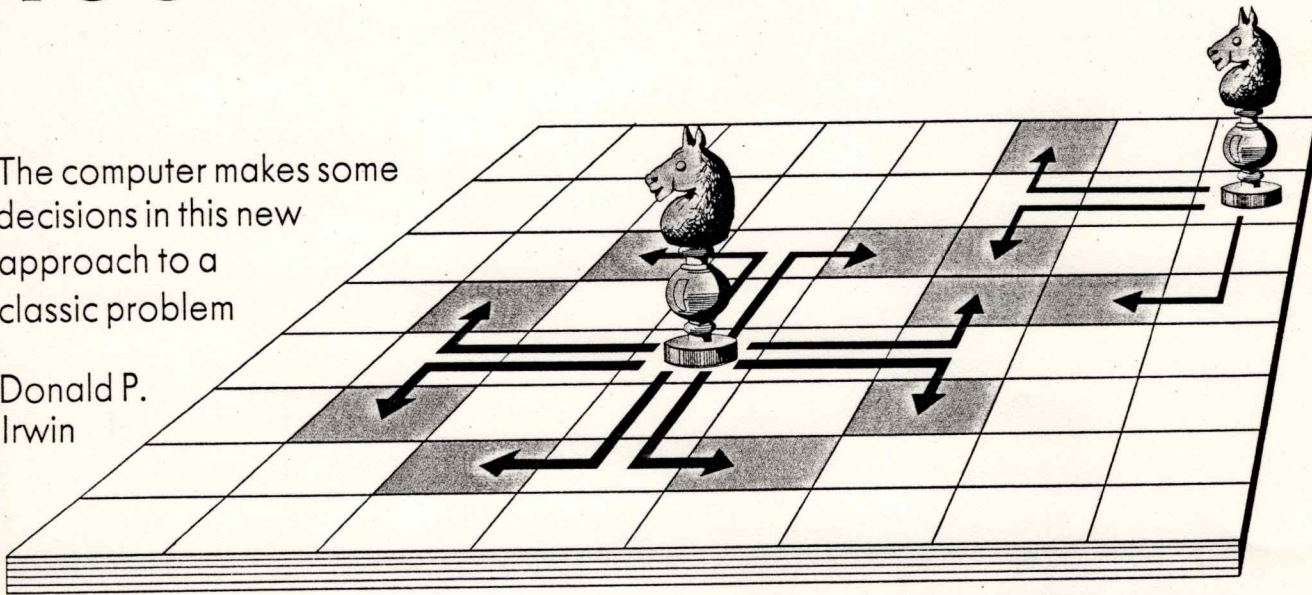


# The Knight's Tour

The computer makes some decisions in this new approach to a classic problem

Donald P. Irwin



Through the ages, fascination with the game of chess and its strategic plays has encouraged many types of complicated and analytical puzzles. One of these is the Knight's Tour, which over the past two centuries has interested some of the greatest minds in the fields of both mathematics and game solutions.

The problem involves the placement of a chess knight anywhere on a standard size chessboard, consisting of 64 squares in an 8x8 configuration, then moving the knight from square to square until all 64 squares have been visited once and only once. The legal move for a chess knight is two squares in a vertical or horizontal direction, then one square perpendicular to the previous square (as shown in Figure 1).

Upon being introduced to this puzzle, your first attempt at a solution might consist of placing the knight on the chessboard and arbitrarily moving it about the board. Even if you have some notion of how to move the knight around the board, you'll find that after two or three tries certain problem areas appear. If these problem areas are not eliminated as soon as possible, they may develop into problems that will eventually terminate the knight's tour.

An example of this arbitrary movement method is shown in Figure 2. The

knight's itinerary is represented by the numbers 1 to 34 with the letter K representing the knight's present position. The square labeled with the letter C indicates a probable trouble area. A trouble area is a square that has only one entry point and one exit point. If this square is not traversed when the knight lands on any of the adjacent squares, as indicated by the arrows in Figure 2, the square will end up as a termination square. A termination square is one that has only one entry point and no exit square. In the example, squares A and B are terminator squares. Obviously, if more than one of these termination squares is produced in a tour, the tour cannot be completed.

This arbitrary trial and error method will eventually produce a solution to the problem, but only after numerous tries and much backtracking. But because there is no guarantee that the itinerary will be completed from any given point, mathematicians and puzzle enthusiasts alike have tried to find different ways in which a solution could always be reached no matter where the knight starts its path.

### Early Approaches

In the early 18th century, people like De Moivre, Euler, Legendre, Roget,

Figure 1. Legal moves for the knight from two positions.

Vandermonde, and Warnsdorff devised some very artistic and practical solutions to this fascinating problem. For an interesting array of these solutions, consult W. W. R. Ball's *Mathematical Recreations and Essays* (Macmillan and Company, Ltd., NY, 1905).

In many of the approaches discussed in that book, the author found extravagant solutions to the Knight's Tour. For example, some people were in-

3	13	11	17	B	A	32	19
10	16	4	1	18	15	30	
5	2	14	12	5	20	33	
15	9	6		34	29	C	
7			K	28	24	21	
					22	27	
					25		23
							26

Figure 2. Example of trouble spots: A and B will terminate the tour.

terested only in creating tours that would be both re-entrant and symmetrical in composition. A re-entrant tour is one in which the last square visited by the knight can lead back to the initial square in only one move. An example of this tour, which is not symmetrical in form, is shown in Figure 3.

Mathematicians were also interested in finding some arithmetic relation between the numbered squares of the completed route. Some were looking for odd and even relationships between adjacent squares or a constant sum, like 260, of squares in each row or column. This latter scheme results in a semi-magic square, in which the diagonals add up to a different sum; no solution to date has provided a perfect magic square and a complete tour of the chessboard at the same time.

J.C. Warnsdorff, a German mathematician, came close to generating all of the possible 31 million solutions. His method, called the Warnsdorff Rule or double-look-ahead, states that the knight should proceed to the square from which the number of available squares during the next two moves is the smallest. Although this rule, developed in 1823, has never been proven accurate, neither has an exception ever been found. The Warnsdorff Rule, more than any other, is useful in both locating the problem areas and providing a way for the knight to eliminate them in advance.

Computing the numbers, to be used for comparison in Warnsdorff's Rule, can be accomplished in four steps. The first step is to find each available square that is adjacent to the knight's present position; these squares are labelled with the letters I, L, and X in Figure 4.

The second step is to count the available squares that can be visited from

the squares labelled in step 1; all of the squares thus "adjacent" to the square labelled I are marked with the letter O. So the total number of squares is seven, as indicated by the subscript on the letter I in Figure 4. The square that the knight currently occupies is not counted.

The third step is to count the squares that can be reached from the O squares as noted in the subscripts.

The fourth step is to add up all the numbers calculated in steps 2 and 3; thus, the number computed for the I

**The algorithm I developed to generate the many solutions to this puzzle uses part of Warnsdorff's Rule along with the arbitrary movement method.**

square would be  $7+8+6+4+3+3+4+6$  for a total of 41. The number for the L square would be  $5+8+8+6+3+4$  or 34. After each of the numbers has been calculated for the remaining X labelled squares, the knight moves to the square possessing the smallest value. In this manner, Warnsdorff was able to see in advance the probable trouble areas.

With the invention of the computer, a whole new breed of people became interested in the Knight's Tour, a problem that lends itself to computer solution. Both new attempts and such modified ones as Professor Bhairav Joshi's

modification of Warnsdorff's Rule (*Creative Computing*, August 1980) have emerged over the last few years. These algorithms have successfully completed the knight's itinerary, but they have neglected the ability of the computer to semi-simulate the human brain.

Past attempts by computer programmers have imposed many restrictions on the movement of the knight; each algorithm could produce only 64 different tours because there are only 64 possible starting squares on the chessboard. However, if you compare the paths generated by different algorithms, you will find that they are totally different.

By incorporating many restrictions in their algorithms, programmers have eliminated the ability of the computer to choose the knight's next move. An example of a program that determines the knight's path before it leaves its initial position is Joshi's modification of Warnsdorff's Rule. In his algorithm, Joshi decided it was better to tell the computer where to move the knight, than to allow the computer the choice of moving the knight into an incomplete tour. Warnsdorff's Rule, however, does allow the algorithm to choose the knight's next move—if there is a tie between the smallest numbers computed. Though it was never proven that this random choice could endanger a solution, Joshi decided it was better to leave it out, thereby limiting the number of routes to 64.

**A New Approach**

Even if the arbitrary choice were installed in the program, it would still not consider all the possible tours. The reason for this is twofold: first there are the restrictions of examining two moves in

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

Figure 3. Example of a re-entrant tour. Location 64 is one move away from location 1.

			O <sub>4</sub>		O <sub>3</sub>		
			O <sub>6</sub>				O <sub>3</sub>
			X		I <sub>7</sub>		
		X	O <sub>8</sub>			X	O <sub>4</sub>
		U <sub>8</sub>		K		O <sub>6</sub>	
	U <sub>6</sub>	X			U <sub>8</sub>	X	
			L <sub>5</sub>		X		
	U <sub>3</sub>				U <sub>4</sub>		

Figure 4. Example illustrates Warnsdorff's Rule for I and L squares.

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

Figure 5. Squares of the chessboard are numbered from left to right.

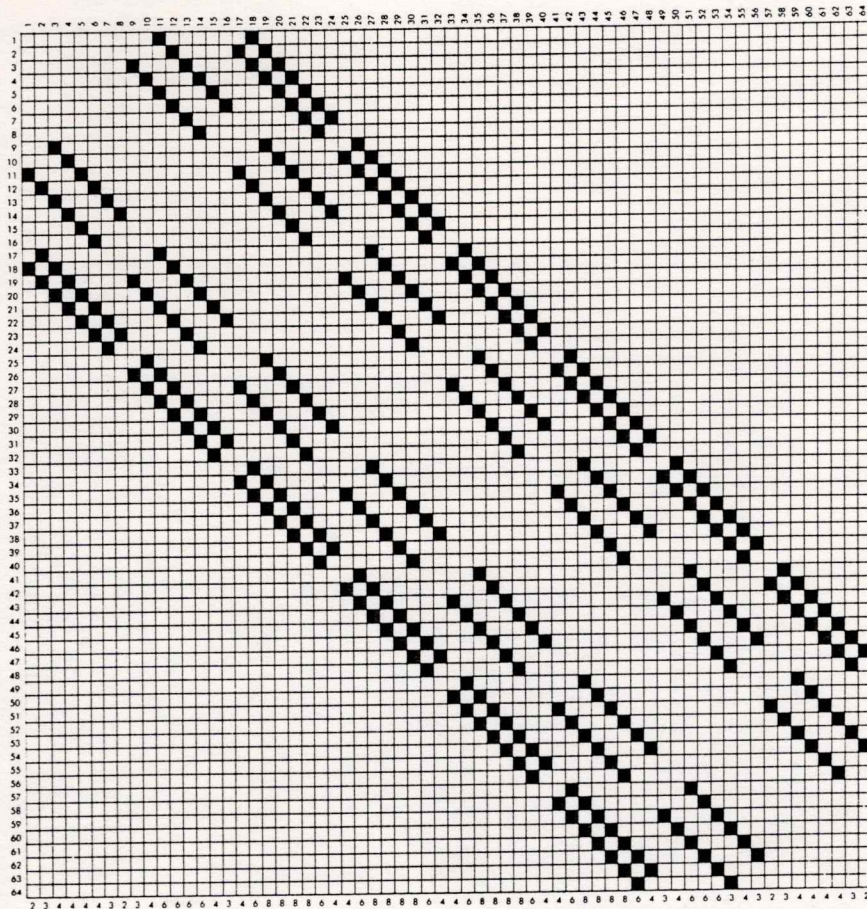


Figure 6. Tabular Form shows legal moves while column totals represent the exit value of each square.

advance, and second, the number associated with the second level adjacent squares is fixed from the start. However, the method presented here makes only one restriction and then allows the computer to make the final decision between moves. There are several methods the computer can use to make arbitrary choices between moves. The most commonly used method, and the one I have incorporated in my program, is the random number generator. Although this method alone can never simulate the indecisiveness of the human brain, it can come close if the generator is truly random in its selection of numbers.

The algorithm I developed to generate the many solutions to this puzzle, uses part of Warnsdorff's Rule along with the arbitrary movement method. I decided that looking ahead, as in Warnsdorff's Rule, was important but not as important as Joshi made it.

By looking so far in advance, Joshi guaranteed success but at the same time greatly reduced the chances of having to make an arbitrary decision. Therefore, I

decided to look at only those squares adjacent to the knight's current position. From them the algorithm picks the one with the smallest number of adjacent squares. Not only does this ensure the elimination of problem areas, but it also increases the probability that the computer will have to make a decision, thereby, producing an efficient algorithm that can generate an almost unlimited number of solutions to the Knight's Tour.

Before presenting my algorithm, I must first define and explain four terms needed to ensure the proper results. First, the chessboard (CB) is an 8x8 set of squares, each of which is labelled from left to right, starting with the number 1 in the upper lefthand corner (see Figure 5).

Second, a set that will represent the squares of the newly defined CB is Let  $S = (1, 2, 3, \dots, 64)$ . Given set S, I can define the binary relation R on S such that (a,b) is in R if and only if there is a legal move from a to b. For example, the pair (1,11) is in R, while the pair (1,5) is not in

R. Since the pair (11,1) is also in R, the binary relation R is defined as a symmetrical relation on S. This symmetrical relation permits a simpler determination of the knight's next move. The tabular form, presented in Figure 6, is the basis of the algorithm's computation.

Third, the exit value (EV) of each square on the CB is the total number of legal moves from that square at a given instance during the tour. This EV is produced by scanning the column associated with a square in the tabular form and counting each of the X's found as shown in Figure 6. The EV determines the knight's next move in the tour.

The process of choosing the correct square is as follows: 1.) find all of the legal squares by scanning down the appropriate column, finding the square or squares with the smallest EV. 2.) if more than one small EV exists, make an arbitrary choice. It should be noted that the reason for picking the smallest EV in step 1 is the same as described in Warnsdorff's Rule. The arbitrary choice in step 2 is what produces the different tours with my algorithm.

In summary, the algorithm is as follows:

*Step 1:* Create a 64 x 64 matrix and initialize it by placing a 1 wherever an X appears in the tabular form in Figure 6 and a 0 in the remaining squares. Create the array EV to store the 64 values described above and initialize it by using the values shown in Figure 6. Create an array CB to store the knight's moves and create the variable KMC to keep track of the knight's movement. Initialize KMC to 1.

*Step 2:* Zero out the matrix row corresponding to the knight's present position (KMC) and subtract 1 from every EV in which a 1 was found in the corresponding column. The reason for this subtraction is to avoid the counting of 1's in each column after every move.

*Step 3:* Examine the column referred to by the knight's present position (KMC) and find the available square with the smallest EV. If more than one square exists, use a random number generator to determine the next move.

*Step 4:* Increment the KMC by 1 and mark the proper location in the CB array with the new value.

*Step 5:* Repeat steps 2 through 4 until the KMC is equal to 64 (a complete tour has been found), or there are no more possible squares to which to move.

*Step 6:* Print out the CB array, which now contains the knight's current path through the chessboard.



**Please let us know 8 weeks in advance so you won't miss a single issue of CREATIVE COMPUTING.**

Attach your old label where indicated and *print* your new address in the space provided. Please be sure to include your mailing label whenever you write us about your subscription. It helps us to serve you promptly.

**Write to:** P.O. Box 5214, Boulder, CO 80322

- Change address only
- Extend subscription
- Enter new subscription—  
one year \$19.97
- Payment enclosed
- Bill me later

**AFFIX OLD LABEL**

If you have no label handy, *print* old address here.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_

- Charge:  American Express  
 Visa  MasterCard

Number \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

**NEW ADDRESS HERE**

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip \_\_\_\_\_

Add \$5 per year in Canada; \$10 per year all other foreign countries. Please allow 60 to 90 days for delivery of first issue.

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

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

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

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

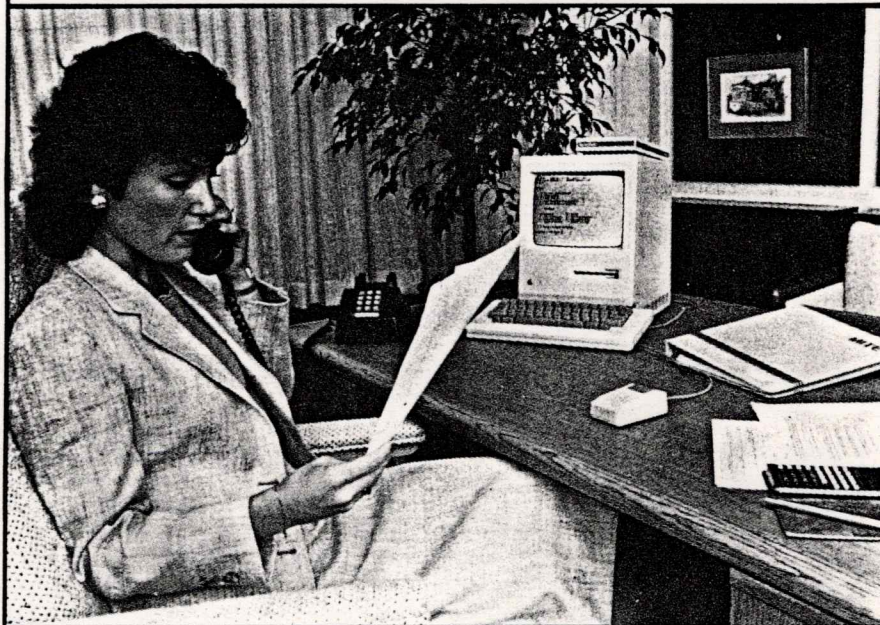
37	2	19	24	35	26	17	14
20	23	36	1	18	15	34	27
3	38	21	62	25	60	13	16
22	55	40	57	64	47	28	33
39	4	63	48	61	32	59	12
54	7	56	41	58	49	46	29
5	42	9	52	31	44	11	50
8	53	6	43	10	51	30	45

**Figure 7. Shows several completed tours from the same starting square. The program located 64 different tours from starting square 12.**

I tested my algorithm on an old IBM 1130 computer using an outdated version of Fortran IV. I generated 20 different itineraries in three seconds of CPU time using the number generator, as compared to 4.946 seconds by Professor Joshi on an Intel AS/6 computer in APL.

I then generated 64 different tours from the same starting square in 9.6 seconds. This is compared to Professor Joshi's time of 6 seconds for 64 itineraries from different starting points. The number of different tours that may be generated and tested by my algorithm depends on the random number generator of the system. The memory of the computer should be able to store the calculations of each new tour generated for comparison with the other tours previously created by the algorithm. ■

# MITE is Changing the Way Businesses Talk!



## Is the Sales Report Ready? Then "MITE It To Me!"

With MITE, the industry's most powerful and easy-to-use data communications program, the world of on-line information is as close as your computer. With the power of MITE, you can automatically log-on to your favorite information service, and capture the information for later editing or analysis. Why bother with overnight express delivery services when you can send documents anywhere in the world in seconds. With MITE, you have the choice of several different error checking protocols including the industry

standard XMODEM protocol with CRC and Batch file capabilities. With the addition of the MITE + Terminal Emulation Module, you can turn your computer into any of over ninety different terminals including a DEC VT100, IBM 3101 or Televideo 950.

MITE is available for over 130 different computer systems (CP/M and MSDOS), including the IBM PC/XT/AT, AT&T 6300, Sanyo 555, Apple Macintosh, DEC Rainbow, Kaypro and Televideo.

Physical Address:  
2615 North Monroe Street  
Tallahassee, FL 32303

**MYCROFT  
LABS INC**

Mailing: P.O. Box 6045  
Tallahassee, FL 32314  
Telephone (904) 385-1141