A SIMPLE ALGORITHM FOR KNIGHT'S TOURS

SAM GANZFRIED

ADVISOR: PAUL CULL
OREGON STATE UNIVERSITY

ABSTRACT. We investigate Warnsdorff's simple heuristic for finding knight's tours on square chessboards and consider various tiebreaking methods. We then analyze one particular algorithm that is consistent with Warnsdorff's rule and runs in linear time. This algorithm requires less memory than other known algorithms and hence is a more practical rule for humans to follow without the aid of a computer. Finally we complete one case of the proof of the correctness of this algorithm.

1. INTRODUCTION

Chess is a two player game played on a square board. Generally chess is played on a board with 8 rows and 8 columns; however, we will deal with the more general case where the board contains m rows and m columns. For the purposes of this paper we will assume the orientation of a board is fixed so that we can label all the squares. Let column 1 denote the leftmost column and row 1 denote the top row. We denote each square by the ordered pair (i, j), where i is the row number and j is the column number. Each player receives six different types of pieces, each of which has a different set of rules governing its possible moves. In this paper we are specifically concerned with the knight. A knight's move follows an L-shape; it can move either one row over and two columns over, or two rows over and one column over (in either direction). We say that two squares are *adjacent* if a knight can move from one square to the other (we also call the two squares *neighbors*). Formally, (a, b) is adjacent to (c, d) if and only if $1 \leq a, b, c, d \leq m$ and

$$(c, d) \in \{(a-2, b+1), (a-1, b+2), (a+1, b+2), (a+2, b+1), (a+2, b-1), (a+1, b-2), (a-1, b-2), (a-2, b-1)\}$$

Notice that adjacency is a symmetric relation, but is neither reflexive nor transitive. It is clear that there are at most eight squares adjacent to any given square; note that there are only two squares adjacent to the corner squares. Since the orientation of the board is fixed, we can label each of these moves as one of eight possible types. These *move-types* are depicted in the following diagram, where X denotes the position of the knight:

	8		1	
7				2
		X		
6				3
	5		4	

A *knight's path* is a sequence of squares such that all pairs of consecutive squares are adjacent and no square appears in the sequence more than once. We can illustrate a knight's path by labeling each square by the the order of its appearance in the sequence, with dashes denoting squares that do not appear in the sequence. At any point in a knight's path we call a square *visited* if it has already appeared in the sequence and *unvisited* if it has not appeared. In our diagrams this means that visited squares contain a number less than or equal to the

number in the current square, while unvisited squares contain either a number higher than the number in the current square or a dash. We call the number of unvisited squares that are adjacent to a given square the *degree* of the square. A *knight's tour* is a knight's path that includes every square of the board. Here is an example of a knight's tour on an 8x8 board:

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

Finding a knight's tour on a chessboard is a special case of the Hamiltonian-Path problem: determining whether there exists a path such that every vertex is visited exactly once in an arbitrary undirected graph. This is easy to see if we think of a chessboard as a graph where the squares are vertices and there is an edge between two vertices if and only if they are adjacent. We call this a *knight's graph*. Then a chessboard has a knight's tour if and only if the corresponding knight's graph has a hamiltonian path. Determining whether an arbitrary graph has a hamiltonian path is an NP-complete problem, and hence no known efficient (polynomial-time) algorithm exists. On the other hand, it is well-known precisely on which chessboards a knight's tour is possible, and several efficient algorithms (linear in the number of squares) exist for constructing such tours [2, 3, 4, 7]. Specifically, an mxm board has a knight's tour if and only if $m \geq 5$. Thus analysis of knight's tours might provide some insight into how to approach the more difficult Hamiltonian-Path problem.

2. WARNSDORFF'S RULE

In 1823, Warnsdorff [9] proposed a simple heuristic for finding knight's tours.

Warnsdorff's Rule: Always move to an adjacent, unvisited square with minimal degree.

Intuitively this seems like a logical rule to follow, since squares with lower degrees have fewer neighbors and therefore we will have fewer opportunities to visit them in the remainder of the path. It is essential to follow this rule if a square has degree 0, since otherwise we can never visit it. Similarly if we fail to visit a square of degree 1, then we will have only one more opportunity to visit it (and it must be the last square of the tour). It is also true that no tour can deviate from Warnsdorff's Rule in the last several moves, although this is less obvious. The idea here is that if we deviate at an early move we have a greater opportunity to overcome this "mistake," while if we deviate near the end of the path then we have fewer chances to avoid failure. Since this "endgame effect" is central to the intuitive appeal of the heuristic, we will make it more rigorous with the following theorem, which relies on a lemma.

Lemma 2.1. *It is impossible to have three mutually adjacent squares on a chessboard.*

Proof. Suppose three squares are mutually adjacent. Then they must all lie in different rows; otherwise two would be in the same row and could not be adjacent to each other. Similarly, all three squares must lie in different columns. Furthermore, the rows of the squares must be consecutive; otherwise two squares would be in rows that have more than two rows between them and could not be adjacent. Similarly, the columns must be consecutive. Consider the square which lies in the middle row. Suppose this square lies in column j. Then one of the other two squares must lie in either column $j + 1$ or column $j - 1$. In either case this square is exactly one row and one column away from the first square and cannot be adjacent to it. So we have a contradiction, and no three squares can be mutually adjacent. □

Theorem 2.2. *It is impossible for a knight's tour to deviate from Warnsdorff's Rule in the last four moves.*

Proof. Suppose we can deviate from Warnsdorff's Rule at the current move of a path and still produce a tour. In particular, suppose we move to a square of degree y instead of a square of degree x, where $x < y$. Let q denote the current square, s_x denote the square of degree x, and s_y denote the square of degree y. If $x = 0$, then it is impossible to produce a tour since we can never visit s_x. So $x \geq 1$. Therefore, $y \geq 2$. So at least three squares must currently be unvisited; otherwise no square could have degree 2. If exactly three squares are unvisited, then we must have $x = 1$ and $y = 2$. Then s_y must be adjacent to s_x since only three squares remain. Therefore q, s_x, and s_y are mutually adjacent. By the lemma this produces a contradiction. Suppose four squares are unvisited. If $y = 3$ then we have the same contradiction since s_x and s_y must be adjacent. So we must have $y = 2$ and $x = 1$, where s_x and s_y are not adjacent. Let a and b denote the other two squares (which must both be adjacent to s_y). If a and b are adjacent then s_y, a, and b are mutually adjacent and we have a contradiction. So a and b are not adjacent. Since s_x has degree 1 and s_x is not adjacent to s_y, s_x must be adjacent to either a or b. Without loss of generality suppose s_x is adjacent to a. Then s_x can not also be adjacent to b. So b has degree 0 (since s_y is now visited). Therefore, we must visit b now or else it will never be visited. But since b is not adjacent to s_x or a, the path terminates and we cannot complete a tour. So we have a contradiction and no such deviation is possible. □

One might begin to wonder whether there are tours that deviate from Warnsdorff's Rule at all; it is not obvious that such tours exist. Unfortunately, it appears that it is relatively easy to "overcome" deviations from the rule if they occur early enough. Most knight's tours do not follow Warnsdorff's heuristic, and many deviate a large number of times. Below is an example of a tour on an 8x8 board that deviates from the heuristic 20 times, although no deviations occur in the last 14 moves. Squares from which the next move is a deviation are in bold:

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

This dramatic example deals a harsh blow to Warnsdorff's heuristic and forces us to consider the opposite extreme. Is Warnsdorff's Rule simply an intuitively appealing heuristic that is utterly ineffective in actually producing tours? One might begin to wonder whether there even exists a single tour that follows Warnsdorff's Rule on each board; the answer is certainly not obvious. That is the question the remainder of this paper will address.

To start with, Warnsdorff's heuristic is not an algorithm; a starting square is not specified, and no rule is specified in the event that more than one unvisited, adjacent square share the minimal degree. Warnsdorff provided one possible solution himself. In the event that several adjacent squares share the minimal degree, Warnsdorff claimed that "[moving] to any of them indifferently" will result in a tour [1]. While it is not clear what square he intended the tour to start on, it seems reasonable to assume it would start at a corner square, which initially has the smallest degree. Thus we can view Warnsdorff's proposal as an algorithm in which ties are broken at random.

Unfortunately, this randomized algorithm does not always work. Here is an example of a path on an 8x8 board that follows Warnsdorff's algorithm but fails to produce a tour:

1	28	13	46	3	26	39	36
14	43	2	27	48	37	4	25
29	12	47	42	45	40	35	38
–	15	44	49	–	–	24	5
11	30	–	–	41	50	–	34
16	–	18	31	–	–	6	23
19	10	–	–	21	8	33	–
–	17	20	9	32	–	22	7

In a 1996 REU paper Squirrel [8] performed 100 trials of this algorithm for all board sizes from 5 to 400; the conclusion was that the algorithm is quite successful for smaller boards, but the success rate drops sharply as board size increases. The algorithm produces a successful tour over 85% of the time on most boards with m less than 50, and it succeeds over 50% of the time on most boards with m less than 100. However, for $m > 200$ the success rate is less than 5%, and for $m > 325$ there were no successes at all. These observations suggest that the success rate of Warnsdorff's random algorithm rapidly goes to 0 as m increases. I confirmed Squirrel's findings, and made another interesting observation. For all boards with $m \leq 25$ the success rate was at least 98% with one surprising exception: for $m = 7$ the success rate was only 75%. I will have more to say about this later in this section.

Despite the failure of Warnsdorff's proposed algorithm, it was believed that Warnsdorff's heuristic could still be salvaged if appropriate improvements were made. Several attempts have been made to improve Warnsdorff's algorithm; however, none of them proved successful on all boards. Parberry [4] considered an algorithm that combined Warnsdorff's heuristic with a random walk algorithm originally proposed by Euler in 1759. Euler's approach was to start at any square, then repeatedly move to a random unvisited neighbor until no more moves are possible. Euler then attempted to replace moves of this path with longer sequences of moves consisting of squares that were not visited by the path. A more complete analysis of Euler's algorithm can be found in [1]. Parberry decided to improve this algorithm by choosing the next square according to Warnsdorff's random algorithm; that is, chose a random square with minimal degree. While an interesting idea, experiments showed that the success rate of the algorithm appears to decrease exponentially and the average running time required to find each tour appears to increase exponentially.

Roth [6] decided that the problem lay in Warnsdorff's random tiebreak rule. He proposed breaking ties by choosing "the successor with largest euclidean distance to the center of the chessboard." It was not comletely clear how he would break ties if more than one square shared the same distance from the center of the board. Roth claimed that his algorithm first failed on a board with 428 rows, and failed less than 1% of the time on all boards with fewer than 2000 rows.

Pohl [5] also considered an alternative to Warnsdorff's tiebreak rule. Instead of breaking ties arbitrarily, Pohl decided to apply Warnsdorff's rule a second time on the squares that share the minimal degree. He proposed taking the sum of the degrees of all the unvisited neighbors of these minimal-degree squares and choosing the square whose sum was minimal. Pohl noted that in theory one could apply this process as many times as necessary to break ties; but he decided that doing so would be computationally inefficient. This is actually not entirely true; for example, if we start in the corner square then any number of iterations of Warnsdorff's rule will result in a tie between the two adjacent squares. In any event, applying Warnsdorff rule a large number of times yields a running time asymptotically equivalent to that of an exhaustive search, as Pohl noted.

I experimented with this second-level Warnsdorff-rule with ties broken at random and made some interesting observations. While Warnsdorff's standard rule succeeded 81% of the time when $m = 50$ and only about 35% of the time for $m = 100$, the second-level rule succeeded over 95% of the time on nearly all boards with $m \leq 100$. I did not have access to the technology to test it on larger values of m, but I suspect the second-level algorithm would eventually converge to a success rate of 0 as well, but for much larger values of m than the first-level algorithm. Also, as in the standard algorithm I noticed one key exception in the data. Every value of m from

5 to 50 had a success rate greater than 97% except $m = 8$, which succeeded on only 67% of the trials. This is very surprising considering that the single-level algorithm succeeded on 99% of the trials for $m = 8$. Applying Warnsdorff's rule a second iteration dramatically decreased the success rate. It would be interesting to see how this pattern continues; for example, would a third iteration of Warnsdorff's rule have a poor performance on a 9x9 board, but a high success rate on 7x7 and 8x8? Unfortunately I did not have time to test this observation in more detail, but I suspect that there would continue to be a single board size with relatively poor performance for each number of iterations. In the 7x7 and 8x8 cases I suspect that the poor performance was due to a single bad decision that is made early on in the path, and I believe a similar phenomenon would continue to occur for some m for a greater number of iterations.

Pohl actually decided not to use a random tiebreak rule if a second application of Warnsdorff's heuristic failed to produce a unique square; he chose to break ties by using a fixed ordering of the squares, similar to our arbitrary ordering of move-types in the introduction. I did not have time to test Pohl's algorithm, but he claims it successfully produces a tour on an 8x8 board. Pohl also proposed several other possible tiebreak solutions. Rather than look at the sum of the squares with minimal degree, we could instead choose the square which has a neighbor with minimal degree. Alternatively, we could use a combination of the two methods, picking squares by minimal sum unless a neighbor of one of the squares has degree 2, in which case we choose that square. Pohl also proposed applying Warndsdorff's standard algorithm, using a sort of backtrack procedure if it fails. Pohl also considered ways of applying Warnsdorff's rule in order to find Hamiltonian paths of more general classes of graphs. Unfortunately most of these ideas were purely speculative, and he did not have any data supporting them.

3. A Successful Warnsdorff-Consistent Algorithm

Both Roth and Pohl introduced the idea of breaking ties according to a fixed ordering of moves rather than using Warnsdorff's random rule. Squirrel decided to investigate this strategy in more detail. Let us define a *move-ordering* as a permutation of the set $\{1,2,3,4,5,6,7,8\}$. We will omit the brackets and punctuation; for example, 28365174 is a move-ordering. If we think of the numbers as corresponding to the move-types presented in the introduction, then each move-ordering provides a deterministic tiebreak rule. For example, if we are using the move-ordering 28365174 and we must choose between breaking a tie with a type-4 move and a type-5 move, we will make the type-5 move because it appears earlier in the move-ordering sequence. The tour presented in the introduction uses Warnsdorff's standard heuristic with the move-ordering 12345678 as a tiebreak rule. Squirrel experimented with move-orderings and concluded that each of the 8! orderings fails on some board size; however, he noticed that certain move-orderings perform better than others in certain situations. Squirrel used Roth and Pohl's approach of breaking ties with move-orderings to develop his own algorithm, which proved to be more effective.

Squirrel's algorithm involves using different move-orderings to break ties depending on the board size and whether we have reached certain squares yet in the path. It is probably easiest to demonstrate the algorithm with an example. In all cases we start in the top left corner and repeatedly apply Warnsdorff's rule; all that changes is the tiebreak rule. If $m \equiv 6 \bmod 8$, then we use the move-ordering 34261578 to break ties until we reach square (6,1). Then we switch to the move-ordering 87642135 until we reach square (3,1). Next we switch to 54132678 until (m-10,1); then 52431678 until (10,m-2); then 85647123 until (3,(m+8)/2). Finally we switch to 12453678, which we use until the end of the path. The full list of move-orderings and switching points for all boards is given on the table on the next page. We will call a square on which the move-ordering changes a *switching square*. Notice that the algorithm consists of eight cases depending on the value of $m \bmod 8$ (actually there are two subcases if $m \equiv 5$), and each case contains five or fewer move-ordering changes. Here is pseudocode for the algorithm given an integer $m \geq 112$:

$m \equiv 0 \bmod 8$	
34261578	$(m-1, m-2)$
87642135	$(2,2)$
51867342	$(m-8, 1)$
51342678	$(7, m-3)$
21435678	end
$m \equiv 1 \bmod 8$	
34261578	$(m-1, m-2)$
87642135	$(2,2)$
51324678	$(m-6, (m+9)/2)$
32481765	end
$m \equiv 2 \bmod 8$	
34261578	$(6,1)$
87642135	$(3,1)$
54132678	$(m-15, 4)$
52431678	$(10, m-2)$
85647123	$(5, (m-6)/2)$
15746823	end
$m \equiv 3 \bmod 8$	
34625718	$(m-1, m-2)$
42681357	$(m-6, m)$
86512347	$(2,5)$
51867342	$(m-10, 3)$
61825437	$((m+1)/2, m-2)$
71642538	end

$m \equiv 4 \bmod 8$	
34261578	$(m-1, m-2)$
87642135	$(2,2)$
51867342	$(m-8, 1)$
51342678	$(10, m-5)$
86753421	$(13, (m+2)/2)$
78563421	end
$m \equiv 5 \bmod 8$	
34261578	$(m-1, m-2)$
87642135	$(2,2)$
51324678	*
15234678	end
$m \equiv 6 \bmod 8$	
34261578	$(6,1)$
87642135	$(3,1)$
54132678	$(m-10, 1)$
52431678	$(10, m-2)$
85647123	$(3, (m+8)/2)$
12453678	end
$m \equiv 7 \bmod 8$	
34625718	$(m-1, m-2)$
42681357	$(m-6, m)$
86512347	$(2,5)$
51867342	$(m-6, 3)$
61825437	$((m+1)/2, m-2)$
61357284	end

* If $m \equiv 5 \bmod 16$, use $(m-2, (m-5)/2)$, else use $(m-2, (m-13)/2)$.

TABLE 1. Summary of tiebreaking rules.

```
A = int[m][m];
K = int[8][2];
int[][] perms;
int[][] switch;

void main() {
    initialize();
    initializeTable();
    Tour();
}

void initialize() {
    for (i = 1; i <= m; i++)
        for (j = 1; j <= m; j++)
            A[i][j] = 0;
    K = [(-2,1),(-1,2),(1,2),(2,1),(2,-1),(1,-2),(-1,-2),(-2,-1)]
}

void initializeTable() - sets the i'th entry of perms to the i'th
    move-ordering and the i'th entry of switch to the i'th switching
    square for the case m mod 8.

void Tour() {
    (x,y) = (1,1);
    (u,v) = switch[1];
    T = perms[1];
    t = 1;
    for (i = 1; (x,y) != (0,0); i++) {
        A[x][y] = i;
        if ((x,y) == (u,v)) {
            t++;
            T = perms[t];
            u = switch[t];
        }
        (x,y) = getNextSquare(x,y,T);
    }
}

int[] getNextSquare(x,y,T) {
    min = 9;
    (a,b) = (0,0);
    for (count = 1; count <= 8; count++) {
        (x',y') = (x,y) + K[count];
        if (isLegal(x',y') && A[x'][y'] == 0){
            deg = degree(x',y');
            if (deg < min) {
```

```
                    min = deg;
                    (a,b) = (x',y');
                    key = count;
               }
               if (deg == min) {
                    for (k = 1; T[k] != key && T[k] != count; k++);
                    if (T[k] == count) {
                         (a,b) = (x',y');
                         key = count;
                    }
               }
            }
        }
        return (a,b);
    }

    boolean isLegal(x,y) {
        return (1 <= x <= m && 1 <= y <= m);
    }

    int degree(x,y) {
        deg = 0;
        for (k = 1; k <= 8; k++) {
            (x',y') = (x,y) + K[k];
            if (isLegal(x',y') && A[x'][y'] == 0)
                deg++;
        }
        return deg;
    }
```

Squirrel claimed that his algorithm holds for all $m \geq 112$. Our experiments reveal that the algorithm actually holds for all m such that $5 \leq m < 112$ as well, with the exception of $m = 74$. Thus we can actually make the stronger statement that it holds for all $m >= 75$. It is uncertain exactly why the algorithm fails for $m = 74$ specifically, but it is not surprising that it does not hold for all small board sizes; on these boards the path runs into the boundaries so early that it cannot follow the complete pattern that makes it successful on larger boards. Our proof of correctness rests on the algorithm following some basic patterns which small enough boards fail to satisfy. Nonetheless, it is easy to test the algorithm with a computer on these boards to observe its success.

Squirrel attempted to prove the correctness of his algorithm for the case $m \equiv 7 \mod 8$. Unfortunately his proof contained several incorrect statements and was not complete. In section 7 we provide a complete proof of this case that holds for all $m \geq 47$. In order to make the patterns on some of the diagrams more clear, we have sometimes made it appear that diagrams contain more than 47 rows; however, this is simply to assist the reader, and some of the rows can be ignored once the pattern is recognized. This will make more sense in section 5 when we introduce form matrix notation.

4. ALGORITHM EFFICIENCY

The running time of Squirrel's algorithm is linear in the number of squares $n = m^2$; initialize() takes $\theta(n)$ time, initializeTable() takes constant time, and Tour() requires n iterations through the while loop, each of which requires a constant number of steps. An astute reader might notice that as implemented the algorithm actually runs in $\theta(nlogn)$, since the numbers that denote the order in which squares are visited actually contain

θ($log n$) bits. However this is a technical detail, and all of the literature I looked at ignored this issue completely. If instead we had chosen to label the squares by the move-type we use to get to the next square, then we only need to deal with numbers from 1 to 8, which require just 3 bits. For the purposes of running time, we can assume that we are using this implementation and that the algorithm runs in linear time.

Squirrel's algorithm is not the first linear time algorithm for finding knight's tours [2, 3, 4, 7]. All of the other algorithms employ a divide and conquer strategy, subdividing the board into several smaller boards and combining tours on these boards into a full tour, using solutions of several small boards as base cases. While these algorithms maintain the same asymtptotic level of efficiency as Squirrel's algorithm and are perhaps more elegant, Squirrel's algorithm is simpler than all of them in the following sense. Technically all of the divide and conquer algorithms as well as Squirrel's algorithm require θ(n) space – as we need a way to access all n squares. In the above pseudocode this takes the form of the array A, which has n entries. However, in addition to the space needed to store the squares Squirrel's algorithm requires just constant space – a few integers and the table of move-orderings and switching squares. The divide and conquer algorithms, on the other hand, require an implicit stack for the recursive calls and use θ($log n$) additional space. Hence if we ignored the space needed to store the squares, Squirrel's algorithm would use much less memory. If we consider the case of a human tracing out the algorithm on an actual chessboard, we can treat the space required by the board as a given for all the algorithms. In this case, Squirrel's algorithm would require asymptotically less space than the other algorithms – meaning the human would need access to less information to complete the tour. Thus Squirrel's algorithm is better suited for actual human implementation than the divide and conquer methods.

5. Form Matrix Notation

Because our proof of the algorithm's correctness is constructive, we need a method of notation that allows us to represent paths on an infinite family of boards of unbounded dimension in a single diagram. For this reason we present a special kind of matrix which we will call a *form matrix*. A form matrix contains five different types of symbols: numbers, lines, dashes, letters without subscripts, and letters with subscripts.

Lines play the role of ellipses and consist of two types: horizontal and vertical. A horizontal line signifies that some number of rows are being omitted at the current position, and a vertical line signifies that some number of columns are being omitted. In order to specify exactly which rows or columns are omitted we will always label the row or column directly before and after each line. Lines are what enable us to represent arbitrarily large matrices in a compact form.

The letter 'x' is a special letter and denotes squares that have been already been visited prior to the current path; we can think of the current path as the concluding subsequence of a longer path with the assumption that the first portion of the path is also a valid knight's path. When we refer to 'letters' with reference to form matrices, we will exclude x. Dashes denote squares that have not been visited previously and are not visited by the current path.

All other symbols denote squares that are visited in the current path. The only numbers we will use are positive integers. If we are currently in a square labeled with a number n, then we use the following rules to determine which square to move to next:

(1) If there is an unvisited, adjacent square labeled with the number $n+1$, move to that square.
(2) If there is at least one unvisited, adjacent square labeled with a letter that either has no subscript or has the subscript '1,' move to the square with these properties that comes first alphabetically.
(3) If no square satisfies either of the previous conditions, terminate the path.

If we are currently in a square labeled with a letter without a subscript, then we use the following rules to determine which square to move to next:

(1) If there is an unvisited, adjacent square labeled with the same letter, move to that square.
(2) If there is an unvisited, adjacent square labeled with the letter that comes next alphabetically with either no subscript or the subscript 1, move to that square.

(3) If there is at least one unvisited, adjacent square labeled with a number, move to the square with these properties that has the smallest number.

(4) If no square satisfies either of the previous conditions, terminate the path.

If we are currently in a square labeled with a letter with a subscript i, then we use the following rules to determine which square to move to next:

(1) If there is an unvisited, adjacent square labeled with the same letter and subscript $i + 1$, move to that square.

(2) If there is an unvisited, adjacent square labeled with the same letter and subscript 1, move to that square.

(3) If there is an unvisited, adjacent square labeled with the letter that comes next alphabetically with either no subscript or the subscript 1, move to that square.

(4) If there is at least one unvisited, adjacent square labeled with a number, move to the square with these properties that has the smallest number.

(5) If no square satisfies either of the previous conditions, terminate the path.

In order to make paths easier to follow, we adopt certain conventions. First, we will always label paths so that they can be followed deterministically by the rules given above. For example, we will never be in a square labeled with an a that is adjacent to two unvisited a's. We will also adopt the following convention with numbers: the first sequence of numbers in a given diagram will start with 1, the next with 100, then 200, 300, etc. This means that when we get to a letter that is not adjacent to the same letter or the letter that comes next alphabetically, we need only look for the number 1 or a multiple of 100. We also label the first and last square of the path in boldface, and also add a star in the first square. Generally we will label squares with letters when they are part of a repeated pattern that will be extended across a line, while numbers will denote clumps of squares that do not extend across a line. Letters with subscripts will be used to label more complicated sequences of letters that are part of a repeated pattern extending across a line. For example, if there is a "cycle" of three move types – such as 4-2-7 – that extends from one side of the board to another, we will label this sequence with a letter containing the subscripts 1, 2, and 3.

We recommend that the reader trace out the path defined by Figure 1 as an exercise before moving on to the next section.

6. METHOD OF PROOF

In the next section we will provide a constructive proof of the correctness of Squirrel's algorithm for the case $m \equiv 7 \bmod 8$ for $m \geq 47$. Because of the messy and somewhat unusual nature of the proof, we use this section to give a preview of the method that we will employ as well as provide a justification of its validity. Our proof consists of a series of steps, each of which presumes the correctness of the previous steps. Each step contains a statement of the form:

> *At some point in the algorithm we will arrive at square (i, j) and exactly the squares that have previously been visited or satisfy at least one of the following conditions will have been visited:*
> *Condition 1*
> *Condition 2*
> *⋮*
> *Condition n*

The statement will be accompanied by a diagram that shows how to construct a path starting from the last square of the previous diagram and ending at the square mentioned in the claim. An explanation of the diagram will also be provided if necessary. Notice that the proof has an inherently recursive structure in that each stage refers back to the previous stages.

A rigorous proof of the above statement requires verifying the following facts:

(1) The path in the diagram is deterministic: applying the rules from the previous section determine a unique path from the start square to the end square.

(2) The diagram starts and ends at the correct squares.

(3) It is obvious how the pattern continues across lines.

(4) Such repeated patterns across lines are in accordance with the row and column labels.

(5) Row and column labels are correct.

(6) A square is visited by the path in the diagram if and only if it satisfies at least one of the conditions of the claim.

(7) A square is marked with an x in the diagram if and only if it has been visited in a previous step of the proof.

(8) A square is marked with a dash in the diagram if and only if it has not been visited in a previous step and it does not satisfy any of the conditions of the claim.

(9) Every move in the path specified by the diagram is in accordance with the algorithm.

It is clear that a form matrix by itself does not constitute a complete proof of the claim; each of the above steps must be justified in order for the proof to be made rigorous. However, we feel that this degree of rigor is not necessary for the proof to succeed; in most cases it is trivial to verify each of these facts by simple inspection of the form matrix, and we assume the reader can do this on his own. Whenever there is a complication in a diagram that makes one of these facts nontrivial, we provide a detailed explanation. However, we still want to encourage the reader who has any doubts on a step of the proof to verify each of the nine statements given above until he has convinced himself of its validity.

7. PROOF OF ALGORITHM CORRECTNESS

Claim 7.1. *At some point in the algorithm we will be in square* $(2,5)$ *and exactly the squares* (i, j) *that satisfy at least one of the following conditions will have been visited:*

(1) $i = 1$, $j \equiv 2$ *or* $j \equiv 3 \bmod 4$

(2) $i = 2$, $j \equiv 0$ *or* $j \equiv 1 \bmod 4$

(3) $j = 1$, $i \equiv 1$ *or* $i \equiv 2 \bmod 4$

(4) $j = 2$, $i \equiv 0$ *or* $i \equiv 3 \bmod 4$

(5) $i = m - 1$, $j \equiv 0$ *or* $j \equiv 3 \bmod 4$

(6) $i = m$, $j \equiv 1$ *or* $j \equiv 2 \bmod 4$

(7) i *odd*, $j = m - 1$ *or* $j = m$

(8) $(i, j) \in \{(2,3), (3,1), (m-2,2), (m,3), (m-5,m), (m-4,m-2), (m-3,m-1), (m-3,m)$
      $(m-2,m-2), (m-1,m-2), (m-1,m-1), (m,m-3)\}$

Proof. Follow Figure 1 from $(1,1)$ to $(2,5)$.   □

Claim 7.2. *For all* $k \equiv 1 \bmod 8$ *such that* $17 \leq k \leq m - 6$ *we will arrive at square* $(k, 3)$ *and exactly the squares that were previously visited or satisfy one of the following conditions will have been visited:*

(1) $i \leq k - 2j + 4$

(2) $i = k - 2j + 6$

(3) $i = k - 2j + 7$ *and* $i \geq 8$

Proof. The base case $k = 17$ can be verified in Figure 2a by extending the path in the previous claim to $(17,3)$, which contains the number '86.' Suppose the conditions hold for $k = p$ where $p \leq m - 14$. Figure 2b shows how to extend the path so conditions are satisfied for $k = p + 8$. The new path follows the sequence 1-a-b-c-d-100-e-f-g-h. The induction terminates when $k = m - 6$ and we arrive at square $(m - 6, 3)$, where the move-ordering changes. We have now filled about $1/4$ of the board.   □

Claim 7.3. *We will reach square* $(1, \alpha)$*, where* $\alpha = (m + 27)/2$ *and exactly the squares which have been previously visited or satisfy at least one of the following conditions will have been visited:*

(1) $i \leq m - 2j + 5$

(2) $i \leq m - 2j + 6$ and $j \leq 8$

(3) $j \leq \alpha - 2i$

(4) $j = \alpha - 2i + 2$

(5) $j = \alpha - 2i + 3$ and $j \geq (m+9)/2$

(6) $(i,j) \in \{(m,4),(m-2,5),(m-1,5),(m-4,6),(m-3,6),(m-2,6),(m,7)\}$

Proof. Follow path in Figure 3 from $(m-6,3)$ to $(1,\alpha)$. This path follows the sequence a-b-c-1-d-100-e-200. $\qquad \square$

Claim 7.4. *Take α as in the preceding claim. Then for all $k \equiv 1$ mod 4 such that $\alpha \leq k \leq m-2$, we will arrive at square $(1,k)$ and exactly the squares which have been previously visited or satisfy one of the following conditions will have been visited:*

(1) $j \leq k - 2i$

(2) $j = k - 2i + 2$

(3) $j = k - 2i + 3$ and $j \geq (m+9)/2$

Proof. The base case $k = \alpha$ was shown in the previous claim. Suppose the conditions hold for $k = p$, where $p \leq m - 6$. Figure 4 shows how to extend the path so the conditions for $k = p + 4$ are satisfied; however, it is not obvious how to label the rows and columns. Notice that the leftmost 'a' occurs at the smallest column j that satisfies both of the following constraints: $i > m - 2j + 5$ and $j = (p+4) - 2i - 3$. Solving these simultaneously yields $j > (2m - p + 9)/3$. Since columns can only take integer values, it must be true that $j \geq (2m - p + 10)/3$. Since p is odd, the a's along the diagonal occur only in even numbered-columns. This means that the leftmost a is in column:

$$C_a = 2 \left\lceil \frac{\left\lceil \frac{2m-p+10}{3} \right\rceil}{2} \right\rceil$$

Similar logic shows that the leftmost a is in row:

$$R_a = \left\lfloor \frac{2p - m - 4}{3} \right\rfloor$$

Notice that there is a single "hump" in the path of c's in the middle of the diagram. This corresponds to the point along the trajectory of the third condition of the inductive hypothesis when the inequality stops holding. The last c before the hump occurs in the same column as the leftmost square that satisfies condition 3 of the inductive hypothesis: hence it lies in column $C_c = (m+9)/2$. The row of the leftmost square satisfying condition 3 of the inductive hypothesis satisfies the following equations: $j = p - 2i + 3$ and $j = (m+9)/2$. The solution is $i = (2p - m - 3)/4$, which is an integer since $p \equiv 1$ mod 4 and $m \equiv 3$ mod 4. Since the last square before the hump is two rows below this square, it lies in row $R_c = (2p - m + 5)/4$. We can now determine all rows and columns of the Figure in terms of p and m. Eventually the induction terminates when $k = m - 2$ and we arrive at square $(1, m-2)$. We have filled another triangular region at the top center of the board, and about $1/3$ of the entire board. $\qquad \square$

Claim 7.5. *We will reach square $(8, m-2)$ and exactly the squares which have been previously visited or satisfy one of the following conditions will have been visited:*

(1) $j \leq m - 2i + 6$

(2) $j = m - 2i + 8$

(3) $j = m - 2i + 9$, $j \geq (m+9)/2$

(4) $(i,j) \in \{(9,m-5),(7,m-4),(5,m-3),(8,m-3),(8,m-2),(7,m-2),(6,m-2),$
$\qquad (6,m-1),(4,m-1),(6,m),(8,m)\}$

Proof. Follow the path in Figure 5 from $(1, m-2)$ to $(8, m-2)$. The path follows the sequence 1-a-b-100-c-d-e-f-g-h-200. Notice the single hump in the trajectory of c's and g's that occurs in the middle of the board.

The rows and columns can be labeled using the same logic as in the previous claim, where the leftmost a is in square (R_a, C_a) and the leftmost c prior to the hump is in square (R_c, C_c):

$$
\begin{aligned}
C_a &= 2\left\lceil \frac{\lceil m/3+4 \rceil}{2} \right\rceil \\
R_a &= \left\lfloor \frac{m-8}{3} \right\rfloor \\
C_c &= \frac{m+9}{2} \\
R_c &= \frac{m+1}{4}
\end{aligned}
$$

$\square$

Claim 7.6. *For all even k such that $8 \leq k \leq (m+1)/2$, we will arrive at square $(k, m-2)$ and exactly the squares which have been previously visited or satisfy one of the following conditions will have been visited:*

(1) $j \leq m - 2i + 2k - 10$
(2) $j = m - 2i + 2k - 8$
(3) $j = m - 2i + 2k - 7$, $j \geq (m+9)/2$
(4) $(i,j) \in \{(k+1, m-5), (k-1, m-4), (k-3, m-3), (k, m-3), (k, m-2), (k-1, m-2)$
$(k-2, m-2), (k-2, m-1), (k-4, m-1), (k-2, m), (k, m)\}$

Proof. The base case $k = 8$ was shown in the previous claim. Now suppose the conditions hold for $k = p$, where $p \leq (m-3)/2$. Figure 6 shows how to extend the path so the conditions for $k = p+2$ are satisfied. We compute the row and column labels as in the preceding claims, once again noting the c-hump:

$$
\begin{aligned}
C_a &= 2\left\lceil \frac{\left\lceil \frac{m-2p+19}{3} \right\rceil}{2} \right\rceil \\
R_a &= \left\lfloor \frac{m+4p-24}{3} \right\rfloor \\
C_c &= \frac{m+9}{2} \\
R_c &= \frac{m+4p-15}{4}
\end{aligned}
$$

One last issue we must contend with is labeling the squares in the two rightmost columns, since some of them were visited in the first step of the algorithm. Since $m \equiv 7 \bmod 8$ and p is even, the quantity $m + 4p - 15$ is divisible by 8, and hence R_c is even. Unfortunately, we cannot determine whether R_a is even, and hence cannot tell whether the squares in the last two columns at the bottom of the diagram have been visited. This is only a minor notational inconvenience, however, as these squares are not relevant to this step of the proof, and for any given m it can easily be computed whether R_a is even or odd. The induction ends at square $(\frac{m+1}{2}, m-2)$, where the move-ordering changes for the last time. All that is left is a region in the bottom right corner of the board. We have now filled about 3/4 of the board. $\square$

Claim 7.7. *We will reach square (β, m), where $\beta = (m+9)/2$ and exactly the squares which have been previously visited or satisfy at least one of the following conditions will have been visited:*

(1) $j \leq 2m - 2i - 2$
(2) $j = 2m - 2i + 1$
(3) $j \leq 12$
(4) $j = 13$ *and* $i \neq m - 3$
(5) $j = 14$ *and* $i \neq m - 5$
(6) $j = 15$ *and* $i \neq m - 5, m - 4,$ *or* $m - 2$
(7) $j = 16$ *and* $i \neq m - 7, m - 6,$ *or* $m - 4$
(8) $(i,j) \in \{(m-7, 17), (m-2, 17), (m-1, 17), (m-9, 18), (m-4, 18), (m-2, 18),$
$(m-1, 18), (m-3, 19), (m-2, 19), (m, 19), (m-3, 20), (m, 20), (m-2, 21),$
$(m-1, 21), (m-2, 22), (m, 23)$
(9) $(i,j) \in \{(\beta-4, m-1), (\beta-2, m-3), (\beta-2, m), (\beta-1, m-2), (\beta, m-4), (\beta, m)\}$

Proof. Follow the path in Figure 7 from $(\frac{m+1}{2}, m-2)$ to (β, m) using the sequence a-1-b-c-100-d-e-f-g-h. As usual, we compute row and column labels:

$$\begin{aligned} C_c &= \tfrac{3m-9}{4} \\ R_c &= \tfrac{m+3}{2} \end{aligned}$$

There are several differences between this path and the previous paths. First, the c-hump takes the form of a type-1 move instead of a type-7 move. Also, the d-trajectory changes direction in the middle of the diagram as we change to e's. More interestingly, note the formations of the f's and g's. The cycle $f_1 - f_2 - f_3 - f_4$ repeats itself for a while until we switch to g's in the middle of the diagram. Then we have a new cycle $g_1 - g_2 - g_3 - g_4 - g_5 - g_6$ that repeats itself until we approach the rightmost boundary of the board. In these cases each f_i and g_i next move to $f_{i+1 \bmod 4}$ and $g_{i+1 \bmod 6}$ respectively if a square with such a labelling is adjacent to the current position. This formation ends when we move from g_5 to an h at the top right of the diagram. It can also be noted that we can label the far right columns since we know the behavior at the bottom rows of the board and R_c is odd. $\qquad\square$

Claim 7.8. *We will reach square (γ, m), where $\gamma = (m+13)/2$ and exactly the squares which have been previously visited or satisfy at least one of the following conditions will have been visited:*

(1) $j \leq 2m - 2i + 2$
(2) $j = 2m - 2i + 5$
(3) $j \leq 20$
(4) $j = 21$ *and* $i \neq m - 7$
(5) $j = 22$ *and* $i \neq m - 5, m - 8,$ *or* $m - 9$
(6) $j = 23$ *and* $i \neq m - 6, m - 7, m - 8,$ *or* $m - 10$
(7) $(i, j) \in \{(m, 24), (m-2, 24), (m-3, 24), (m-5, 24), (m-1, 25), (m-2, 25), (m-3, 25),$
$\qquad (m-4, 25), (m-1, 26), (m-2, 26), (m-4, 26), (m, 27), (m-2, 27), (m-3, 27),$
$\qquad (m, 28), (m-3, 28), (m-1, 29), (m-2, 29), (m-2, 30), (m, 31)\}$
(8) $(i, j) \in \{(\gamma, m), (\gamma-4, m-1), (\gamma-4, m-2), (\gamma-2, m-2), (\gamma-1, m-2), (\gamma-2, m-3),$
$\qquad (\gamma, m-3), (\gamma-3, m-4), (\gamma, m-4), (\gamma-1, m-5)\}$

Proof. Follow the path in Figure 8 using the sequence a-1-b-100. The formation of the b's is identical to that of the g's in the previous diagram. $\qquad\square$

Claim 7.9. *We will reach square (θ, m), where $\theta = (m+29)/2$, and exactly the squares which have been previously visited or satisfy at least one of the following conditions will have been visited:*

(1) $j \leq 2m - 2i + 18$
(2) $j = 2m - 2i + 21$
(3) $j \leq 28$
(4) $j = 29$ *and* $i \neq m - 3$
(5) $j = 30$ *and* $i \neq m - 1, m - 4,$ *or* $m - 5$
(6) $(i, j) \in \{(\theta, m-4), (\theta, m-3), (\theta, m), (\theta-1, m-5), (\theta-1, m-2), (\theta-2, m-3),$
$\qquad (\theta-2, m-2), (\theta-2, m), (\theta-3, m-4), (\theta-4, m-2), (\theta-4, m-1)\}$

Proof. Follow the path in Figure 9 using the sequence:

$$a - 1 - b - 100 - c - 200 - d - 300 - e - 400 - f - 500 - g - 600 - h - 700$$

While the diagram seems quite complicated, it is really only four repetitions of the pattern we saw in the last two diagrams. The trajectories of a,c,e, and g consists of type-6 moves (with no humps), while the trajectories of b,d,f, and h consist of the 6-cycles we saw previously. $\qquad\square$

Claim 7.10. *Take θ as in the preceding claim. Then for all $k \equiv \theta \bmod 6$ such that $\theta \leq k \leq m-9$, we will arrive at square (k, m) and exactly the squares that have been previously visited or satisfy at least one of the following conditions will have been visited:*

(1) $j \leq m - 2i + 2k - 11$
(2) $j = m - 2i + 2k - 8$
(3) $j \leq 28 + 2(k - \theta)$
(4) $j = 29 + 2(k - \theta)$ *and* $i \neq m - 3$
(5) $j = 30 + 2(k - \theta)$ *and* $i \neq m - 1, m - 4,$ *or* $m - 5$
(6) $j = 31 + 2(k - \theta)$ *and* $i = m$
(7) $(i, j) \in \{(k, m-4), (k, m-3), (k, m), (k-1, m-5), (k-1, m-2), (k-2, m-3),$
$(k-2, m-2), (k-2, m), (k-3, m-4), (k-4, m-2), (k-4, m-1)\}$

Proof. The base case $k = \theta$ was shown in previous claim (notice square $(m, 31)$ was actually visited in step 8). Suppose the conditions hold for $k = p$ where $p \leq m - 15$. Figure 10 shows how to extend the path so that the conditions hold for $k = p + 6$. The path follows the following sequence:

$$a - 1 - b - 100 - c - 200 - d - 300 - e - 400 - f - 500$$

The trajectories of the letters are identical to those in Figure 9. Notice we will end at a different square (in terms of m) depending on the value of m mod 6. If $m \equiv 1$ mod 6 we end at $(m - 13, m)$, if $m \equiv 3$ mod 6 we end at $(m - 11, m)$, and if $m \equiv 5$ mod 6 we end at $(m - 9, m)$. In all three cases we have filled the entire board except for a few hundred squares. $\square$

Figure 11 shows how to complete the tour in each of the three cases. Figure 11a shows the case $m \equiv 1$ mod 6, Figure 11b shows the case $m \equiv 3$ mod 6 and Figure 11c shows the case $m \equiv 5$ mod 6. When $m \equiv 1$ or 5 mod 6, the tour will end at square $(m - 4, m - 8)$ while if $m \equiv 3$ mod 6, the tour ends at $(m - 6, m - 4)$.

8. CONCLUSION

Unfortunately it appears that the proofs of the other cases will be just as lengthy, and I do not have enough time in the REU program to prove these cases. While I cannot present a complete proof of the algorithm at this time, I did test the algorithm on boards up $m = 610$ and confirmed success on all of these for $m \geq 75$. This strongly suggests that the algorithm is successful on all larger boards as well. Additionally, it does not appear that there is anything special about the case $m \equiv 7$ mod 8, and I suspect that the other cases can be proved by similar arguments. Figure 12 provides an overview of the paths produced by applying the algorithm in the different cases. The path first visits the region marked 1, then 2, etc. These diagrams should provide some clues as to how the proofs should proceed in the other cases.

Assuming the proof of correctness can be completed, this algorithm would provide an efficient way of generating knight's tours on all boards with $m \geq 5$ except $m = 74$. If $m = 74$, then applying Warnsdorff's Rule using the move-ordering 21345678 to break ties produces a successful tour. Thus, an algorithm for finding a tour on all square boards with $m \geq 5$ that is consistent with Warnsdorff's Rule would be to use Squirrel's algorithm if $m \neq 74$, and use the move-ordering 21345678 if $m = 74$. While perhaps not as simple and elegant as Warnsdorff might have wanted, this is the only known algorithm for finding knight's tours that is consistent with Warnsdorff's heuristic. As the discussion in section 4 indicates, the algorithm maintains a linear running time and requires less space than other known algorithms, thus making it easier for humans to apply without the aid of a computer. Probably the biggest accomplishment of the algorithm is that it provides an example of turning a promising but unsuccessful common-sense heuristic into an effective and efficient algorithm. Hopefully this approach can be used by computer scientists to find efficient solutions to a variety of other problems.

APPENDIX

FIGURE 1

	1										11	m-7							m
1	a*	a	c	-	-	a	e	-	-	a	e	-	-	a	e	-	-	a	e
	c	-	a	a	**e**	-	-	a	e	-	-	a	e	-	-	a	e	-	-
	a	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	a
	-	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	e
	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	a
	-	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	e
	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	a
12	-	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
m-10	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	e
	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	-	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	a
	-	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	a	e
	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	d
	-	b	-	-	-	-	-	-	-	-	-	-	-	-	-	-	d	e	a
	-	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	d	d
	b	c	-	-	-	-	-	-	-	-	-	-	-	-	-	-	e	a	a
	c	-	d	b	-	-	d	b	-	-	d	b	-	-	d	b	a	d	d
m	c	b	c	-	d	b	-	-	d	b	-	-	d	b	-	e	d	a	a

FIGURE 2. A

	1										11	m-7							m
1	x	x	x	7	12	x	x	45	78	x	x	-	-	x	x	-	-	x	x
	x	8	x	x	**1***	46	39	x	x	-	-	x	x	-	-	x	x	-	-
	x	x	6	11	38	13	44	77	66	79	-	-	-	-	-	-	-	x	x
	9	x	15	2	47	40	65	42	-	-	-	-	-	-	-	-	-	-	-
	x	5	10	37	14	43	76	67	80	-	-	-	-	-	-	-	-	x	x
	x	16	3	48	21	64	41	-	-	-	-	-	-	-	-	-	-	-	-
	4	x	22	19	36	75	68	81	-	-	-	-	-	-	-	-	-	x	x
	17	x	49	26	63	20	-	74	-	-	-	-	-	-	-	-	-	-	-
	x	23	18	35	50	69	82	-	-	-	-	-	-	-	-	-	-	x	x
10	x	34	25	62	27	-	73	-	-	-	-	-	-	-	-	-	-	-	-
	24	x	32	51	70	83	-	-	-	-	-	-	-	-	-	-	-	x	x
	33	x	61	28	-	72	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	31	52	71	84	-	-	-	-	-	-	-	-	-	-	-	-	x	x
	x	60	29	-	53	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	30	x	58	85	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
	59	x	-	54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	57	**86**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
	x	-	55	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	56	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
20	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
m-6	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x
	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x
	-	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x
	x	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	x	x	x
	x	-	x	x	-	-	x	x	-	-	x	x	-	-	x	x	x	x	x
m	x	x	x	-	x	x	-	-	x	x	-	-	x	x	-	x	x	x	x

FIGURE 2. B

	1								9	$\frac{p-11}{2}$										$\frac{p+9}{2}$
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	a	c	x	x
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	a	c	x	x	-	-
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	b	d	e	g	-
	x	x	x	x	x	x	x	x	x	x	x	x	x	a	c	e	g	-	-	-
	x	x	x	x	x	x	x	x	x	x	x	x	x	x	b	d	f	h	-	-
	x	x	x	x	x	x	x	x	x	x	x	x	a	c	e	g	-	-	-	-
	x	x	x	x	x	x	x	x	x	x	x	x	x	b	d	f	h	-	-	-
	x	x	x	x	x	x	x	x	x	x	x	a	x	e	c	-	g	-	-	-
	x	x	x	x	x	x	x	x	x	x	x	x	b	d	f	h	-	-	-	-
	x	x	x	x	x	x	x	x	x	x	a	x	e	c	-	g	-	-	-	-
	x	x	x	x	x	x	x	x	x	x	x	b	d	f	h	-	-	-	-	-
	x	x	x	x	x	x	x	x	x	a	x	e	c	-	g	-	-	-	-	-
	x	x	x	x	x	x	x	x	x	x	b	d	f	h	-	-	-	-	-	-
p-9	x	x	x	x	x	x	a	x	e	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	x	x	x	b	d	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	x	a	x	e	c	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	x	x	b	d	f	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	a	x	e	c	-	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	x	b	d	f	h	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	a	x	e	c	-	g	-	-	-	-	-	-	-	-	-	-	-
	x	x	x	x	b	d	f	h	-	-	-	-	-	-	-	-	-	-	-	-
	x	x	a	x	e	c	-	g	-	-	-	-	-	-	-	-	-	-	-	-
p	x	x	1*	b	d	f	h	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	a	x	e	2	-	g	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	x	7	d	f	h	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	8	x	e	3	-	g	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	6	d	f	h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	e	4	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	5	x	105	h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	106	x	-	101	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	104	**h**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	x	-	102	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	103	x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
p+11	-	x	-	h	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

FIGURE 3

FIGURE 4

FIGURE 5

FIGURE 6

FIGURE 7

FIGURE 8

FIGURE 9

FIGURE 10

FIGURE 11. A

```
        m-27              m-20                              m-10                                        m
m-16 | x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   3   x   x   x
     | x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   x   x   x   x   x  85   4   x   x   x   2   x
     | x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   x   x   x  84   5   x   x  86  99   x   x   x
     | x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   x  83   6   x  90 125 100   x   x  87  98  1*
     | x  x  x  x  x  x  x   x   x   x   x   x   x   x  78   7   x  91 124 101  94  89 158 133  96   x   x
     | x  x  x  x  x  x  x   x   x   x   x   x   x  77   8   x  82 119 102  93 126 157 134  95  88 159 132  97
     | x  x  x  x  x  x  x   x   x   x   x  44   9   x  79 118 103  92 123 152 135 128 218 160 189 130   x   x
     | x  x  x  x  x  x  x   x   x  43  10   x  76 117 104  81 120 151 136 127 156 238 190 129 217 161 188 131
     | x  x  x  x  x  x  x  42  11   x  45 116 105  80 113 150 137 122 153 242 191 219 216 225 187 182   x   x
     | x  x  x  x  x  41  12   x  38  53 106  75 114 149 138 121 146 241 192 155 239 232 237 220 215 226 162 183
     | x  x  x  32 13  x  39  52  71  46 115 108 139 112 147 210 193 154 240 235 243 221 224 227 181 186   x   x
     | x  x  14  x  40 35 48  37  54 107  74 111 148 141 194 145 240 211 244 222 231 236 233 215 172 163 184   x
     | x  x  33 16 31 26 51  72  47  70  55 140 109 144 207 196 209 204 230 **246** 234 223 228 180 185   x   x   x
     | x  15  x 19 34 49 36  25  56  73 110  67  60 195 142 199 206 245 212 203 229 179 214 171 164 173   x   x
     | x  x  x 30 17 20 27  50  69  24  59  64 143  66  61 208 197 200 205 178 213 170 165 174 167   x   x   x
     | x  x  18  x  x 29 22   x  57  68   x   x  63 198   x   x 177 202   x   x 175 168   x   x   x   x
   m | x  x  x  x 21  x  x  28  23   x   x  58  65   x   x  62 201   x   x 176 169   x   x 166   x   x   x
```

FIGURE 11. B

```
        m-23                                        m-10                                  m
m-14 | x  x  x  x  x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   x   3   x   x   x
     | x  x  x  x  x  x  x  x  x  x  x   x   x   x   x   x   x   x  77   4   x   x   x   2   x
     | x  x  x  x  x  x  x  x  x  x  x   x   x   x   x  76   5   x   x  78  91   x   x   x
     | x  x  x  x  x  x  x  x  x  x  x   x   x  75   6   x  82 109  92   x   x  79  90  1*
     | x  x  x  x  x  x  x  x  x  x  x  42   7   x  83 108  93  86  81 174 131  88   x   x
     | x  x  x  x  x  x  x  x  x 41  8   x  74 107  94  85 110 171 132  87  80 175 130  89
     | x  x  x  x  x  x  x 40  9   x  43 106  95  84 103 166 133 112 173 170 181 114   x   x
     | x  x  x  x  x 39 10  x 36 51 96  73 104 155 134 111 172 167 182 113 176 129 180 115
     | x  x  x 30 11  x 37 50 69 44 105 98 135 102 165 156 161 150 169 **184** 179 116   x   x
     | x  x  12  x 38 33 46 35 52 97  72 101 154 137 160 149 168 183 162 145 128 177 118   x
     | x  x  31 14 29 24 49 70 45 68  53 136  99 140 153 164 157 148 151 178 117   x   x   x
     | x  13  x 17 32 47 34 23 54 71 100  65  58 159 138 141 152 163 144 127 146 119   x   x
     | x  x  x 28 15 18 25 48 67 22  57  62 139  64  59 158 143 126 147 120 123   x   x   x
     | x  x  16  x  x 27 20  x  x 55  66   x   x  61 142   x   x 121 124   x   x   x   x   x
   m | x  x  x  x 19  x  x 26 21  x   x  56  63   x   x  60 125   x   x 122   x   x   x   x
```

FIGURE 11. C

```
        m-19                          m-10                            m
m-12 | x  x  x  x  x  x  x  x   x   x   x   x   x   x   x   x   3   x   x
     | x  x  x  x  x  x  x  x   x   x   x   x   x   x  47   4   x   x   x   2   x
     | x  x  x  x  x  x  x  x   x   x   x  46   5   x   x  48  81   x   x   x
     | x  x  x  x  x  x  x  x   x  45   6   x  92 113  80   x   x  49  78  1*
     | x  x  x  x  x  x  x 44   7   x  91 118 105  88  93 112  79  82   x   x
     | x  x  x  x  x 43 14  x  40 117 104  89 114 119 124  87  94  77  50  83
     | x  x  x 34 15  x 41 70 103  90 115 122 131 106 111  76 125  86   x   x
     | x  x  16  x 42 37 68 39 116 101 132 107 120 123 130  95  60  51  84   x
     | x  x  35 18 33 28 71 102 69 108 121 **134** 129 110  75 126  85   x   x   x
     | x  17  x 21 36 67 38 27  72 133 100 109  74 127  96  59  52  61   x   x
     | x  x  32 19 22 29 66 99  26  73 128  97  58  53  62  55   x   x   x
     | x  x  20  x  x 31 24  x   x  65  98   x   x  63  56   x   x   x   x
   m | x  x  x  x 23  x  x 30  25   x   x  64  57   x   x  54   x   x   x   x
```
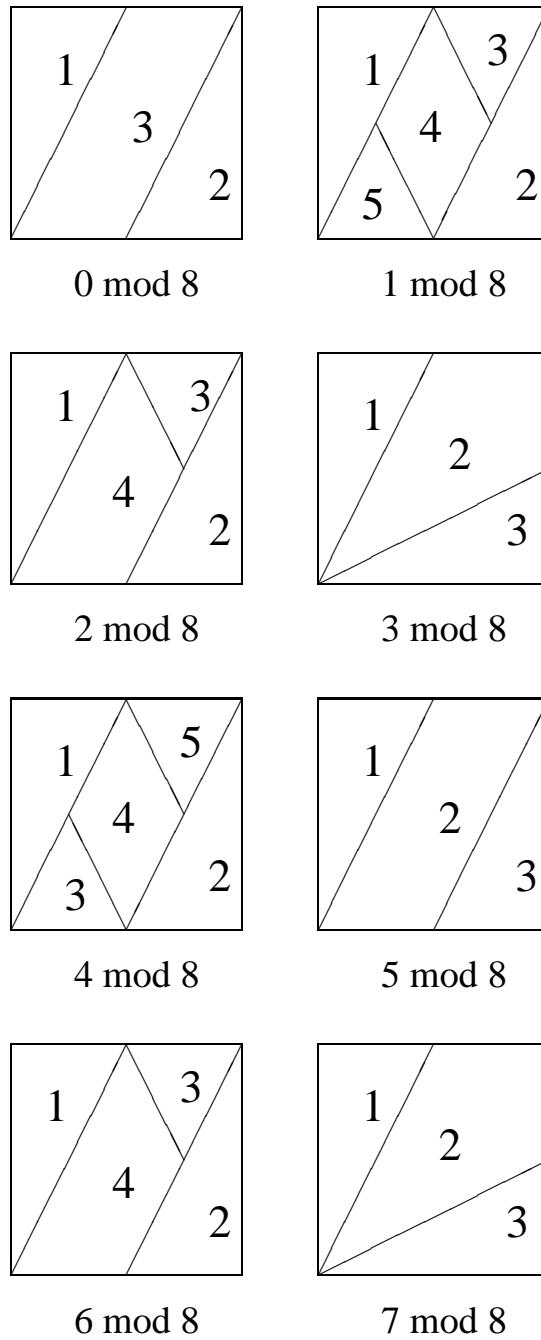
FIGURE 12. Summary of tours produced by algorithm.

References

[1] Ball, W.W. Rouse and H.S.M. Coxeter. *Mathematical Recreations and Essays*. 12th ed. Toronto: University of Toronto, 1974.

[2] Conrad, A., Tanja Hindrichs, Hussein Morsy, and Ingo Wegener. Solution of the knight's Hamiltonian path problem on chessboards. *Discrete Applied Mathematics* 50 (1994) 125-134.

[3] Cull, Paul and Jeffrey De Curtins. Knight's tour revisited. *Fibonacci Quarterly* 16 (1978) 276-285.

[4] Parberry, Ian. Algorithms for touring knights. Technical Report CRPDC-94-7, Center for Research in Parallel and Distributed Computing, Dept. of Computer Science, University of North Texas, May 1994.

[5] Pohl, Ira. A method for finding Hamilton paths and knight's tours. *Communications of the ACM* 10 (1967) 446-449.

[6] Roth, Arnd. The problem of the knight. http://sun0.mpimf-heidelberg.mpg.de/people/roth/Mma/Knight.html

[7] Schwenk, Allen J. Which rectangular chessboards have a knight's tour? *Mathematics Magazine* 64 (1991) 325-332.

[8] Squirrel, Douglas and Paul Cull. A Warnsdorff-Rule Algorithm for Knight's Tours on Square Chessboards. Oregon State REU Program, 1996.

[9] Warnsdorff, H.C. Des Rösselsprunges einfachste und allgemeinste Lösung, Schmalkalden (1823).

Harvard University

E-mail address: ganzfr@fas.harvard.edu