Knights Tour Problem

(Report)

Algorithm Analysis
Project

Sarantos Tzortzis
Vilnius University, 2021

Contents:

1. Introduction……………………………………………………………………….3
2. Problem formulation…………………………………………………………..3
3. Sample of the solution………………………………………………………..4
    3.1 Backtrack ………………………………………………………………….4
    3.2 Warnsdoff…………………………………………………………………..8
4. Implementation………………………………………………………………….9
5. A description of the experiments performed with the samples when the size tended to infinity……………………………………….10
6. Complexity……………………………………………………………………….11
7. Instruction on how to use the program……………………………….12
8. Conclusion and observations…………………………………………….12
9. References……………………………………………………………………….14

1.Introduction

A knight's tour is a sequence of moves of a knight on a chessboard such that the knight visits every square exactly 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. Variations of the knight's tour problem involve chessboards of different sizes than the usual 8 × 8.

2.Problem formulation

The famous knight's tour problem asks whether a knight can tour an entire 8 × 8 chessboard visiting each square exactly once. Here, a knight can move in any of 8 ways, provided that the final destination is within the board. In each of these ways, one coordinate of the knight's position changes by 2 units (positively or negatively), and the other coordinate changes by 1 unit (positively or negatively). If the two directions are labelled up/down and left/right the eight moves are :

• Up two steps, right one step 2
• Up two steps, left one step
• Right two steps, up one step
• Right two steps, down one step
• Down two steps, left one step
• Down two steps, right one step
• Left two steps, up one step
• Left two steps, down one step. From the above listing of moves, it is clear that the knight's move is symmetric, in the sense that if a knight can move from a square A to a square B in one move, it can also move from B to A in one step. There are many variations of this problem :
• Whether a tour is possible with a given initial position.
• Whether a tour is possible with a given initial and a given final position.
• Whether there is a closed tour or re-entrant tour , that is, a tour where the last square is a knight's move away from the first. If there is a closed tour

beginning at some square, there is a closed tour beginning at any square. Essentially, the knight needs to cover as many squares of the chessboard such that after each square, the next square it goes to is a knight's move away. We would like a structure that captures this relationship between squares (of being separated only by a knight's move). Such a structure exists in mathematics – a graph. 1

3.Sample of the solution

3.1 Back-tracking implementation

Here are some examples from my program

```
Enter N
5
Enter Starting position X
0
Enter Starting position Y
0
-------------------------------------
|  0  ||  5  || 14 ||  9  || 20 |
-------------------------------------
| 13 ||  8  || 19 ||  4  || 15 |
-------------------------------------
| 18 ||  1  ||  6  || 21 || 10 |
-------------------------------------
|  7  || 12 || 23 || 16 ||  3  |
-------------------------------------
| 24 || 17 ||  2  || 11 || 22 |
-------------------------------------
```

```
Enter N
5
Enter Starting position X
4
Enter Starting position Y
4
-------------------------------------
| 24 || 17 ||  6  || 11 ||  2  |
-------------------------------------
|  7  || 12 ||  3  || 16 ||  5  |
-------------------------------------
| 20 || 23 || 18 ||  1  || 10 |
-------------------------------------
| 13 ||  8  || 21 ||  4  || 15 |
-------------------------------------
| 22 || 19 || 14 ||  9  ||  0  |
-------------------------------------
```

```
Enter N
6
Enter Starting position X
2
Enter Starting position Y
3
-------------------------------------
| 35 || 12 ||  9  || 22 ||  3  || 14 |
-------------------------------------
| 10 || 21 ||  4  || 13 ||  8  || 23 |
-------------------------------------
| 19 || 34 || 11 ||  0  || 15 ||  2  |
-------------------------------------
| 28 || 31 || 20 ||  5  || 24 ||  7  |
-------------------------------------
| 33 || 18 || 29 || 26 ||  1  || 16 |
-------------------------------------
| 30 || 27 || 32 || 17 ||  6  || 25 |
-------------------------------------
```

```
Enter N
6
Enter Starting position X
1
Enter Starting position Y
1
-------------------------------------
| 18 || 27 || 34 || 25 || 20 ||  5  |
-------------------------------------
| 35 ||  0  || 19 ||  6  || 33 || 24 |
-------------------------------------
| 28 || 17 || 26 || 23 ||  4  || 21 |
-------------------------------------
| 11 || 14 ||  1  || 30 ||  7  || 32 |
-------------------------------------
| 16 || 29 || 12 ||  9  || 22 ||  3  |
-------------------------------------
| 13 || 10 || 15 ||  2  || 31 ||  8  |
-------------------------------------
```

```
Enter N
7
Enter Starting position X
0
Enter Starting position Y
0
-------------------------------------------
|  0  || 37 || 30 ||  7  || 18 || 35 || 14 |
-------------------------------------------
| 31 || 28 || 19 || 36 || 15 ||  6  || 17 |
-------------------------------------------
| 38 ||  1  || 32 || 29 ||  8  || 13 || 34 |
-------------------------------------------
| 27 || 24 || 39 || 20 || 33 || 16 ||  5  |
-------------------------------------------
| 40 || 21 ||  2  || 25 || 44 ||  9  || 12 |
-------------------------------------------
| 23 || 26 || 47 || 42 || 11 ||  4  || 45 |
-------------------------------------------
| 48 || 41 || 22 ||  3  || 46 || 43 || 10 |
-------------------------------------------
```

```
Enter N
7
Enter Starting position X
0
Enter Starting position Y
6
-------------------------------------------------
| 16 || 27 || 42 || 23 ||  8  || 29 ||  0  |
-------------------------------------------------
| 41 || 24 || 15 || 28 ||  1  || 22 ||  9  |
-------------------------------------------------
| 26 || 17 || 40 || 43 || 10 ||  7  || 30 |
-------------------------------------------------
| 39 || 44 || 25 || 14 || 31 ||  2  || 21 |
-------------------------------------------------
| 34 || 47 || 18 || 37 ||  4  || 11 ||  6  |
-------------------------------------------------
| 45 || 38 || 35 || 32 || 13 || 20 ||  3  |
-------------------------------------------------
| 48 || 33 || 46 || 19 || 36 ||  5  || 12 |
-------------------------------------------------
```

```
Enter N
8
Enter Starting position X
0
Enter Starting position Y
0
-------------------------------------------------------
|  0  || 59 || 38 || 33 || 30 || 17 ||  8  || 63 |
-------------------------------------------------------
| 37 || 34 || 31 || 60 ||  9  || 62 || 29 || 16 |
-------------------------------------------------------
| 58 ||  1  || 36 || 39 || 32 || 27 || 18 ||  7  |
-------------------------------------------------------
| 35 || 48 || 41 || 26 || 61 || 10 || 15 || 28 |
-------------------------------------------------------
| 42 || 57 ||  2  || 49 || 40 || 23 ||  6  || 19 |
-------------------------------------------------------
| 47 || 50 || 45 || 54 || 25 || 20 || 11 || 14 |
-------------------------------------------------------
| 56 || 43 || 52 ||  3  || 22 || 13 || 24 ||  5  |
-------------------------------------------------------
| 51 || 46 || 55 || 44 || 53 ||  4  || 21 || 12 |
-------------------------------------------------------
```

3.2 Warnsdoff Implementation

```
PROBLEMS 16    OUTPUT    DEBUG CONSOLE    TERMINAL

Enter N
5
Enter Starting position X
0
Enter Starting position Y
0
0       13      8       19      2
23      18      1       14      9
12      7       24      3       20
17      22      5       10      15
6       11      16      21      4
Sarantoss-MacBook-Pro:KnightsTourWarnsdorff sarantostzortzis$
```

```
Enter N
8
Enter Starting position X
5
Enter Starting position Y
4
5       22      7       36      3       20      17      34
8       63      4       21      56      35      2       19
23      6       55      60      37      18      33      16
62      9       52      57      54      59      38      1
51      24      61      42      47      0       15      32
10      27      48      53      58      41      44      39
25      50      29      12      43      46      31      14
28      11      26      49      30      13      40      45
Sarantoss-MacBook-Pro:KnightsTourWarnsdorff sarantostzortzis$
```

```
Enter N
15
Enter Starting position X
6
Enter Starting position Y
10
42      13      92      95      44      15      152     97      46      17      156     51      48      19      158
91      94      43      14      149     96      45      16      151     216     47      18      157     52      49
12      41      104     93      100     153     150     211     98      155     162     217     50      159     20
103     90      101     148     131     196     99      154     215     220     213     170     161     164     53
40      11      132     105     134     147     210     193     212     171     222     163     218     21      160
89      102     135     130     195     192     197     190     221     214     219     202     169     54      165
10      39      106     133     146     137     194     209     200     207     172     223     166     203     22
79      88      129     136     139     198     191     144     189     224     201     206     173     168     55
38      9       80      107     128     145     138     199     208     183     186     167     204     23      174
71      78      87      74      81      140     127     182     143     188     205     184     177     56      117
8       37      72      77      108     123     0       141     126     185     178     187     116     175     24
65      70      75      86      73      82      109     124     181     142     115     176     179     118     57
36      7       66      69      76      85      122     1       110     125     180     119     114     25      28
67      64      5       34      83      62      3       32      121     60      111     30      27      58      113
6       35      68      63      4       33      84      61      2       31      120     59      112     29      26
Sarantoss-MacBook-Pro:KnightsTourWarnsdorff sarantostzortzis$
```

```
Enter N
19
Enter Starting position X
0
Enter Starting position Y
0
0       3       40      91      86      5       84      93      98      7       96      201     114     9       222     119     116     11      228
41      88      1       4       83      92      99      6       95      200     113     8       221     202     115     10      227     120     117
2       39      90      87      102     85      94      199     112     97      220     203     240     247     232     223     118     229     12
89      42      103     82      107     194     111     100     219     204     239     250     233     224     241     246     231     226     121
38      59      108     195     110     101     218     205     198     215     234     261     248     251     258     225     242     13      230
43      104     81      106     193     206     197     216     235     262     249     238     259     298     245     252     257     122     243
58      37      60      109     196     217     208     263     214     237     260     359     296     293     256     299     244     253     14
47      44      105     80      207     192     213     236     265     350     285     292     319     354     297     294     255     300     123
36      57      46      61      54      209     264     211     284     291     360     349     358     295     320     335     302     15      254
45      48      55      186     79      212     191     266     351     286     345     318     353     330     355     306     321     124     301
56      35      62      53      190     187     210     283     290     317     352     357     348     343     336     329     334     303     16
49      52      173     78      185     176     267     188     269     346     287     344     331     356     307     338     305     322     125
34      63      50      165     174     189     184     177     282     289     316     347     342     337     326     333     328     17      304
51      164     75      172     77      160     175     268     183     270     341     288     325     332     339     308     323     126     311
64      33      140     161     166     171     178     159     168     281     182     315     340     279     324     327     310     277     18
141     74      163     76      143     156     167     170     179     152     271     280     181     314     309     278     129     312     127
32      65      142     139     162     69      144     155     158     169     180     151     272     149     130     313     276     19      22
73      138     67      30      71      136     157     28      145     134     153     26      147     132     273     24      21      128     275
66      31      72      137     68      29      70      135     154     27      146     133     150     25      148     131     274     23      20
Sarantoss-MacBook-Pro:KnightsTourWarnsdorff sarantostzortzis$
```

4.A brief description of the implemented algorithms

4.1 Backtracking

In order to solve Knights Tour Problem, I implemented "solveRec()" which is getting six parameters:

- **x, y** = Position of Knight
- **Chessboard[][]** = Two-dimension array which represents the chessboard
- **moveNumber** = A counter to help us for the move's number in order to fill the array correctly
- **horiMoves, verMoves** = horizontal moves and vertical moves that a knight can do. horiMoves[x] and verMoves[x] is knight's valid move.

This method is using all possible combination of moves from horiMoves and verMoves arrays in order to fill the path. In case that the knight reaches in a block that there are not any other possible moves, and this move is not the last one, we use **backtracking**. The knight is going to the previous block and tries next move (combination) of horiMoves, verMoves array. SolveRec() will return either true or false in case there is or not a solution.

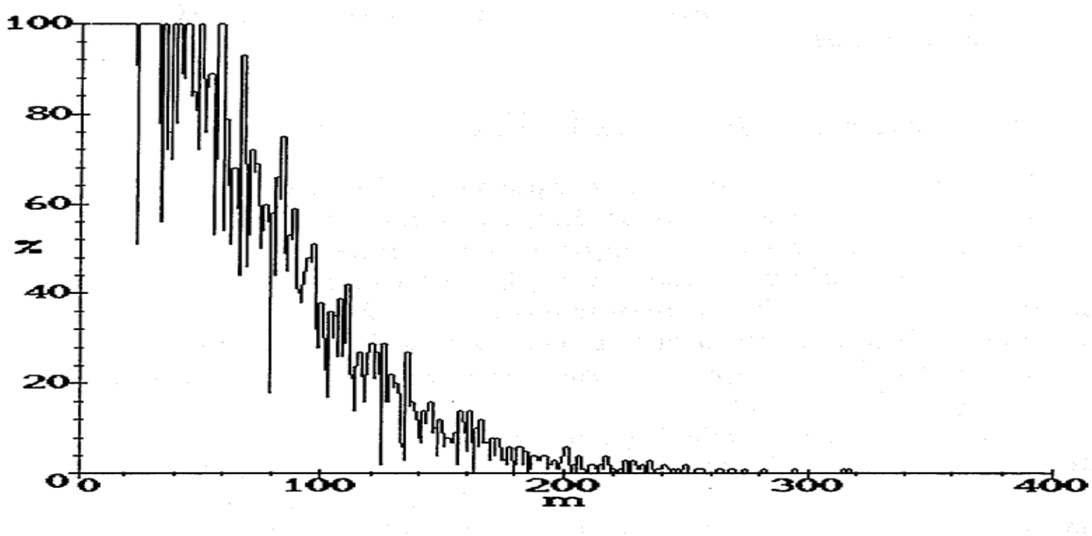
Also I implemented run() which is a method to initialize the chessboard and calls solveRec().

4.2 Warnsdoff

In order to solve Knights Tour Problem using Warnsdoff I created “solve()” function which generates all the legal moves using Warnsdoff heuristic, otherwise returns false. This function initializes an array, which represents our chessboard, with all its values equals to -1. Then it declares the starting position and creates a Tile (Tile is an auxiliary object in order to be easier to represent our tiles on the chessboard as we are not using a two-dimension array on this approach). After that, we are running a for loop in order to find next moves. Then, nextMove function, is trying to find the neighbor with the minimum degree (degree = number of adjacent, unvisited tiles). To do so, we are checking each neighbor and if his degree is less than our minimum, we choose him and we update our points.

5.When the Samples tend to infinity

When the samples tend to infinity using **Warnsdoff rule**, this heuristic is controversial. While its very accurate on small dimension arrays, the success rate tends to be low on bigger ones. On the other hand, the time complexity is still very good as I run the program for N=300 and I got results in less than 10 seconds.

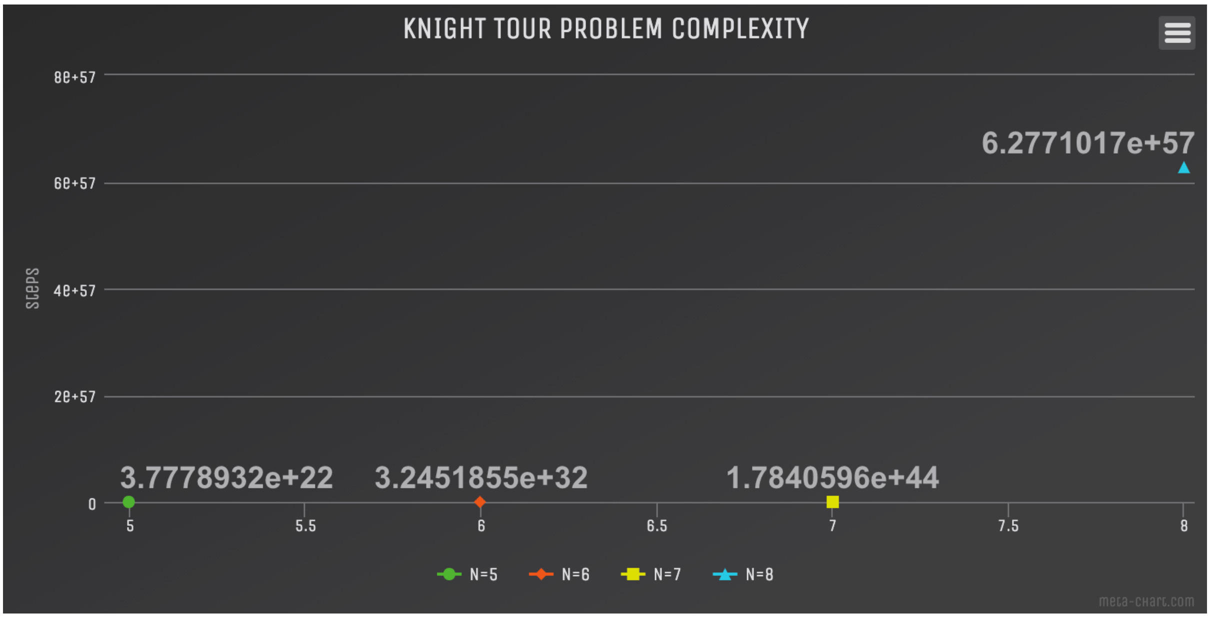


**I was not able to test backtracking approach for chessboards bigger than 8x8 as my device is weak. **

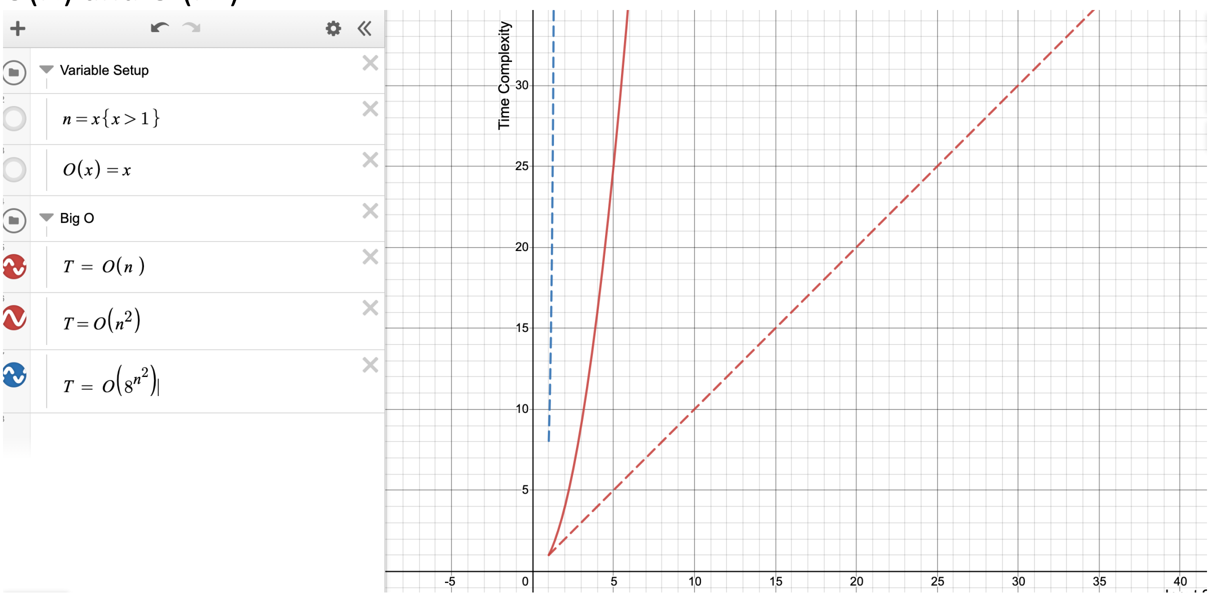
6.Complexity

6.1 Backtracking

Lets consider N the lengths of the chessboard. There are N^2 cells and for each cell, we have a maximum of 8 moves to choose from, so the worst running time is $O(8^{N^2})$.

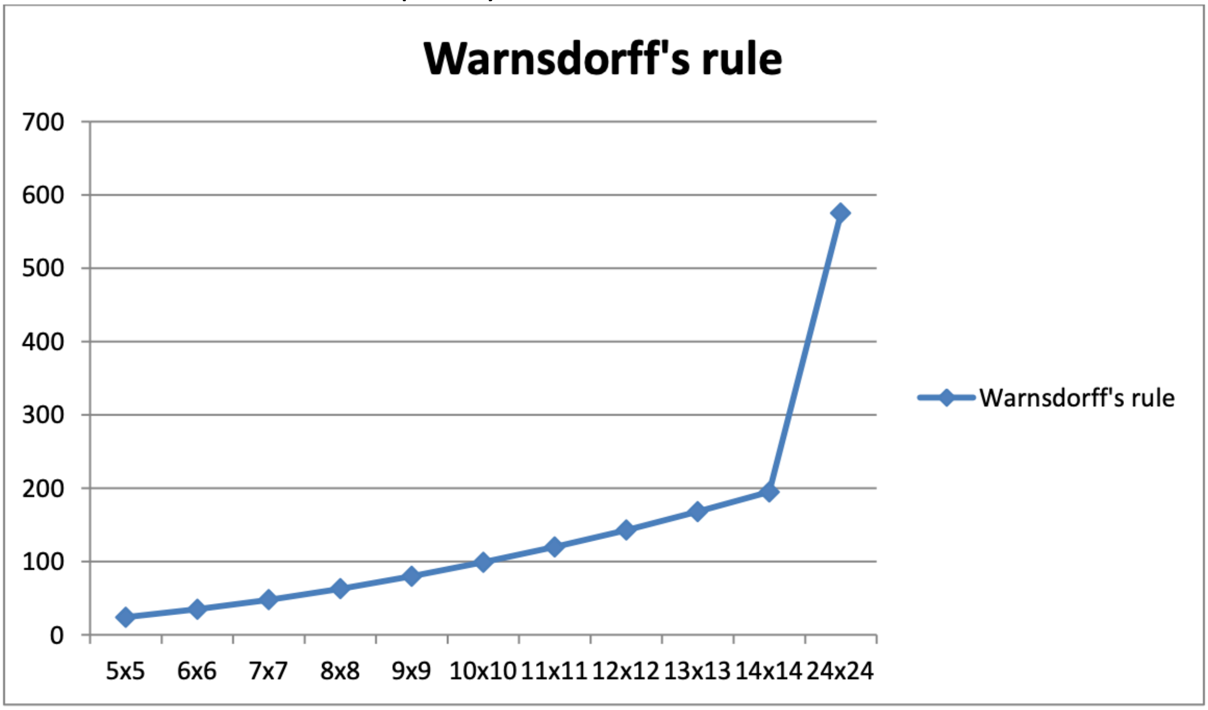


In order to understand how big this complexity is, here is $O(8^{N^2})$ compared to O(N) and O (N^2)



6.2 Warnsdoff's Rule

Warnsdoff rule complexity tend to be linear



7.Instructions

When you download and open the program, it will ask you to input "N" which is the dimensions of the chessboard (NxN). After that, you have to input X, Y which are the points on chessboard that you want the knight to start from. **Values should be in range [0, N-1].**
i.e. N = 5, x=2, y=2

8.Conclusion and observations

After this project I managed to realize how important is to create a fast and optimized algorithm. Knights tour problem can be solved with backtracking using just a two-dimension array, but the complexity is huge. The program, in the worst case, must try every possible move from each tile of the chessboard in order to find a solution. Furthermore, the values on the two arrays that represent the possible moves of the knight is very important. Think of a case, when a person chooses 6 wrong long paths and finally reaching the goal in the 7th path and another case when the person took the correct path in the first turn. You can try this by running the program for N=8 and X, Y=0. The solution will appear immediately (because the two arrays which represent knights moves are optimized for going to the right side of the chessboard). But if you try to run the program with parameters N=8 and X, Y =3,2 it will take some time in order to print the solution because of this movement table optimization.

9.References

Tools used for charts.

Complexity Chart

https://www.desmos.com

Knights Tour Complexity chart

https://www.meta-chart.com

Knights tour - Wikipedia
https://en.wikipedia.org/wiki/Knight%27s_tour

An efficient algorithm for the Knight's tour problem - Ian Parberry * Department of Computer Sciences, University of North Texas
https://www.mimuw.edu.pl/~rytter/TEACHING/ALCOMB/parberry_algoknight.pdf

A Warnsdoff-Rule algorithm for knights' tour in square chessboards – Douglas Squirrel & Paul Cull
http://sites.science.oregonstate.edu/math_reu/proceedings/REU_Proceedings/Proceedings1996/1996Squirrel.pdf

Knight's tour algorithm-Hassan Tariq
https://www.slideshare.net/deadman211/knights-tour-algorithm