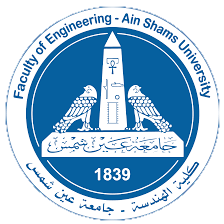
****

Course Name: Design and Analysis of Algorithms

Course Code: CSE332s

**Algorithms Project**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Omar Osama Abdelmonem** | **2001754** |

Team Member

**Task 1**

Assumptions

1. Start position is randomized every time.
2. The algorithm will always find a solution.
3. First Position will not be counted as a move.
4. We always move to an adjacent, unvisited square with minimal degree (minimum number of unvisited adjacent).

Problem Description

A knight is placed on the first block of an empty board and, moving according to the rules of chess, must visit each 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 (or re-entrant); otherwise, it is open. This is a Hamiltonian path which is a NP-hard in general.

On an 8 × 8 board, there are exactly 26,534,728,821,064 directed closed tours (i.e. two tours along the same path that travel in opposite directions are counted separately, as are rotations and reflections). The number of undirected closed tours is half this number, since every tour can be traced in reverse

Pseudo Code & Solution Steps

1. Set a random initial position on the board – P.
2. Mark the board at the initial position with the move number “0”.
3. Repeat the following for the remaining squares in the board:
   1. let S be the set of positions accessible from P.
   2. Set P to be the position in S with minimum accessibility.
   3. Mark the board at P with the current move number.
4. Return the board with each square marked with the move number on which it is visited.

Complexity Analysis

T(N) = O( N^2 Log(N) )

Comparison with another technique

Compared to brute force method, this algorithms archives less time complexity as in brute force the time complexity equals to O( 8 ^ (N^2) ) which is not acceptable. However both algorithms require the same space complexity O (N^2).

Output

Every time the algorithm is set to run, it manages to get a solution for the problem and always return different results but all are success. The algorithm always make constant number of moves to solve the problem; Total number of moves is 63 if first move is not counted as we assumed later – 64 if we counted the first move; That is the number of board squares (8\*8).

Here are 2 examples:

A picture containing text, meter, device

Description automatically generated

A picture containing text, meter, device

Description automatically generated

Conclusion

According to Warnsdorff rule, the problem is managed to be solved in much less time rather than using brute force techniques.

**References**

* Dally, Simon, ed. (1984). *Century/Acorn User Book of Computer Puzzles.*
* Alwan, Karla; Waters, K. (1992). *Finding Re-entrant Knight's Tours on N-by-M Boards* . ACM Southeast Regional Conference.
* Pohl, Ira (July 1967). "A method for finding Hamilton paths and Knight's tours". *Communications of the ACM*.
* Squirrel, Douglas; Cull, P. (1996). ["A Warnsdorff-Rule Algorithm for Knight's Tours on Square Boards"](https://github.com/douglassquirrel/warnsdorff/blob/master/5_Squirrel96.pdf?raw=true).