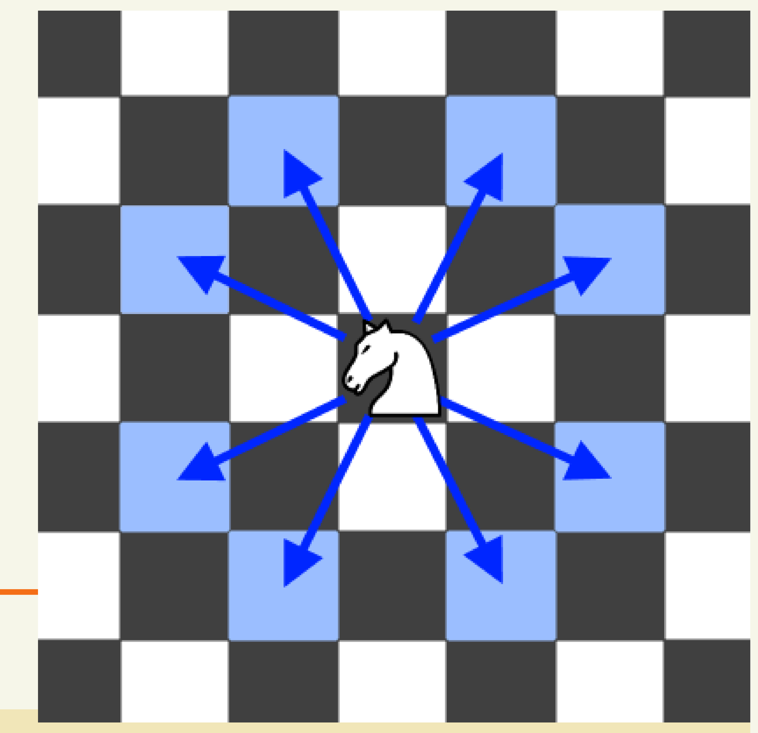
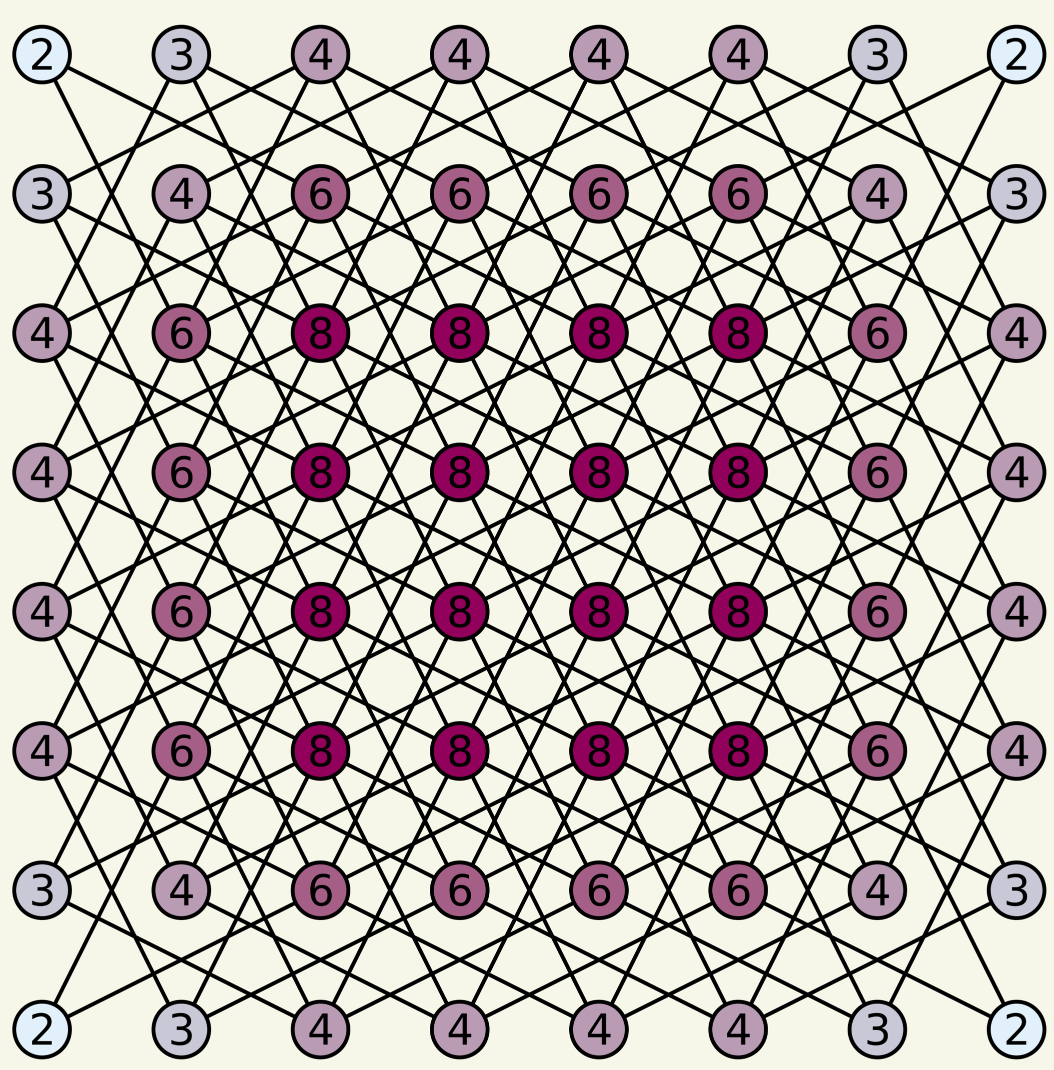
KNIGHT'S TOUR PROBLEM

Given an empty 8x8 chessboard with a knight piece placed on an arbitrary square. Does a sequence of moves exist so the Knight visits each square once? If so, find and display that sequence.



GRAPH THEORY

The chessboard can be converted to a graph as follows:
Each square on the board represents a vertex of the graph, where if there is a possible move for the knight from one square to another, there will also be an edge connecting the vertices. The degree of a vertex is the possible move for the knight from that vertex. The knight's graph below shows the graph and its' vertex



HAMILTON PATH/CYCLE

A Hamilton path is a path in a graph that visits each vertex once. Similarly, a Hamilton cycle is a cycle that visits each vertex once. A Hamiltonian path that starts and ends at adjacent vertices can be completed by adding one more edge to form a Hamiltonian cycle, and conversely, removing any edge from a Hamiltonian cycle produces a Hamiltonian path.

EXPRESS FORMALLY IN MATH

Does a Hamilton cycle exist on the knight's graph? If so, design an algorithm to find it, and analyze it's time and space complexity.

NAIVE ALGORITHM

The naive algorithm generates all possible tours individually and checks if the tour satisfies the constraint. If not, it will backtrack to find another tour. More specifically, it will do as follows:

Step 1. Check if all squares are visited, stop if true.
Step 2. Add one of the next moves to the solution vector, and recursively check if this move leads to a solution.
Step 3. If the move chosen in the above step doesn't lead to a solution, remove this move from the solution vector and try other alternative moves.

COMPLEXITY ANALYSIS

Time Complexity:
In the worst case, a move can take up to 8 iterations. We have n*n squares in total(where n is the dimension of the board). Therefore, O(8 ** (n**2)) is the worst running time.

Space Complexity:
We have n*n squares in total, so the auxiliary space is O(n**2).

WARNSDORFF'S ALGORITHM

This algorithm starts with a knight on any initial position and moves to the next adjacent unvisited vertex with a minimal degree of unvisited vertex. More specifically, it will do as follows:

Step 1. Set P to be any position on the board.

Step 2. Mark P with the current number of move.
Step 3. Let S be the set of positions accessible from P.
Step 4. Set P to be the position in S with minimum accessibility.

Repeat step 2 until the board is fully solved and return.

COMPLEXITY ANALYSIS

Time Complexity:
Each step will take a constant amount of time to search for the smallest degree, and we have n*n squares in total. Therefore, O(n**2) is the worst running time.

Space Complexity:
We have n*n squares in total, so the auxiliary space is O(n**2).