Introduction

Solve the knight's tour problem using backtracking methodology.

The Knight's Tour is a classic chess problem where the goal is to move a knight around the board so that it visits every square exactly once. The knight moves in an L-shape: two squares in one direction and then one square perpendicular, or vice versa. Solving the Knight's Tour involves figuring out a sequence of moves that allows the knight to visit each square without repeating any. Our approach utilizes backtracking, a recursive algorithmic technique that searches for and builds a solution incrementally, backtracking upon reaching a dead-end.

Backtracking is every time you come to a juncture where you have multiple paths to follow, follow the first one and see if it leads to a solution. If it doesn't come back and choose the next solution, which means backtracking is a recursive algorithm.

Instructions on how to run the code

1. Input Dialog

- When you run the code, an Input Dialog will appear.
- In this dialog, you'll need to input the chessboard dimensions (number of rows and columns). By default, the values are set to 3x4, but you can adjust them according to your preference.
- Additionally, you'll find an option to toggle "Show Backtracking". If checked, it will display all the backtracking steps during the solution process.
- Click the "OK" button to proceed.

2. Chessboard Visualization

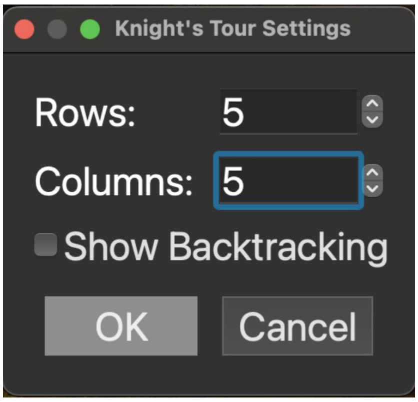
- After clicking "OK," a new window will pop up.
- This window will display the chessboard with the specified dimensions.
- The knight's moves will be shown as a sequence of numbers (representing the order in which cells are visited).
- If you toggle on "Show Backtracking", you'll see the backtracking steps highlighted in "red" color.

3. Solution and Output

- If a solution exists (i.e., a valid knight's tour is found), the console will print "Solution Found."
- The final board will be printed, showing the knight's path.
- If no solution exists (e.g., for a 4x4 board), a Message Box will appear, indicating "No path found."

Required formatting of input

- The chessboard dimensions (number of rows and columns) must be specified as integers.
- The minimum allowed value for both rows and columns is 1, and the maximum allowed value for both rows and columns is 20.



Knight's Tour Settings

Rows: 5

Columns: 5

□ Show Backtracking

Time Complexity

The Knight's Tour algorithm using backtracking has a worst-case time complexity of O(8^(N × M), where N is the number of rows and M is the number of columns on the chessboard. The best-case time complexity is O(N × M) when we find a valid tour quickly without backtracking.

Acknowledgments

[GB17] Debajyoti Ghosh and Uddalak Bhaduri. "A Simple Recursive Backtracking Algorithm for Knight's Tours Puzzle on

Standard 8×8 Chessboard”. In: *IEEE Conference Proceedings* (2017).

[Gee24] GeeksforGeeks. *The Knight’s Tour Problem*. https://www.geeksforgeeks. org/the-knights-tour-problem/. Apr. 2024.

[SKS15] M. Singh, A. Kakkar, and M. Singh. “Image Encryption Scheme Based on Knight’s Tour Problem”. In: *Procedia Computer Science* 70 (2015), pp. 245–250.

[Swe21] Charlotte Sweeney. “The Knight’s Tour: Finding Some of its Solutions”. In: (2021).

[Wik24] Wikipedia. *Knight’s Tour*. https : / / en . wikipedia . org / wiki / Knight’s_tour. Aug. 2024.