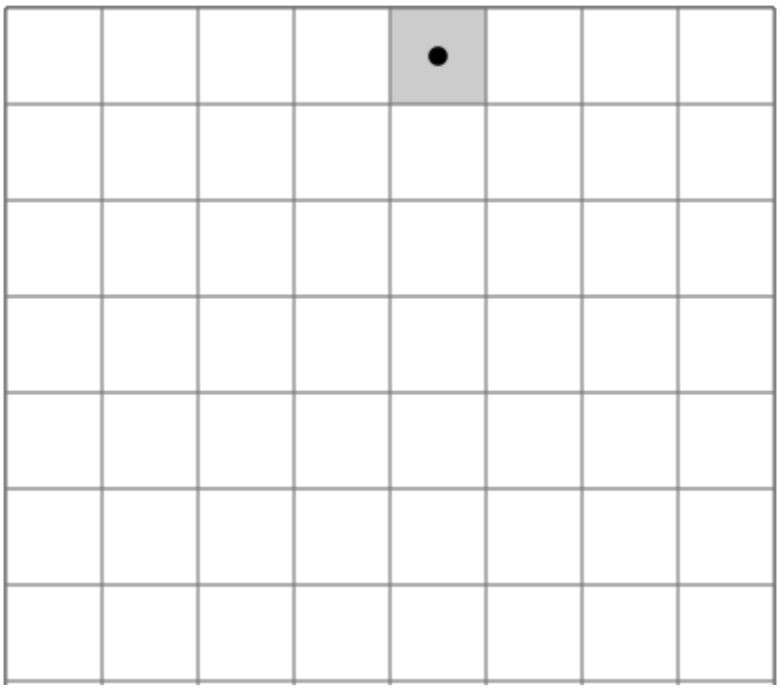
quora.com

(87) Akin Williams's answer to What are some programs every programmer should make at least once?

3 minutes

The Knight's Tour is both a mathematical and logical problem in which a knight visits every square on a chess board of variable size only once and, optionally, returns to the first square after the tour is complete. It is a good way of evaluating algorithm design and efficiency.



Programmers with a classical CS background will have encountered this problem, or one like it, in their education.

As a predominantly self-taught programmer, attempting to solve the Knight's Tour gave me a lot of insight into what programmers should think about when designing applications. It tested my ability to think laterally and devise creative solutions to a real-world problem.

I started with a brute-force approach: considering every square within two moves and then backtracking out of dead-ends. Then I re-evaluated and refined the logic in my tour program to consider more efficient methods of solving the problem on boards of variable sizes. I tried different strategies like making smaller "virtual" boards from bigger ones, ranking move choices, checking efficiency from different starting points, etc.

It was a really fun and cool little experiment.

Update: My tiny mind might have expanded just a bit when I implemented Warnsdorff's Heuristic in my path finding algorithm. At first, it seemed counter-intuitive that searching for positions with the fewest subsequent moves would improve efficiency. That was before I figured out that I had been focusing on "finding" viable paths instead of eliminating unviable ones. Dead-ends are the enemy of my knight's efficiency, so eliminating them at all costs should be the

priority. The following basic algorithm looks for a path with the fewest branches, which statistically eliminates more dead-ends and raises efficiency.

1. Set P to be a random initial position on the board
2. Mark the board at P with the move number “1”
3. Do following for each move number from 2 to the number of squares on 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 marked board — each square will be marked with the move number on which it is visited.

Source: Warnsdorff's algorithm for Knight’s tour problem - GeeksforGeeks

The updated version is on my Github repo for anyone that is interested.