Erik Rodriguez
Ren Hao Wong
Steve Jaimes
Zheng Wei Ng

The N Queens problem

Motivation:

To find a solution to the n-queens problem using a brute force algorithm and to verify a knights move solution in terms of providing complete data of the n-queens problem

Problem Statement:

What is an optimized brute force algorithm? What are some common patterns and problems? How reliable it knights move to get these solutions?

Background Material:

A standard 8×8 board consists of 92 solutions that are made up of 12 fundamental solutions through rotations and reflections. Of the 12 fundamental solutions, it can be observed that the knight's move relationship occurred for 53 times between 2 queens. This makes up for around 4.4166 occurrences per board and an estimated 55.25% occurrence rate. The

Experiment set-up (including metrics, benchmarks, data collection):

Preparations: A board is generated and queens are placed, When a queen is placed it updates all effected tiles and then the next queen is placed

Measurement: The successful board layouts of queens are added to a list of solutions

Procedure: The board dimensions for the problem were first initialized. The board dimensions may be transposed for algorithm compatibility to ensure: rows <= columns. NQueens() or NQueens_KnightsMove() will then be called as a recursive backtracking algorithm to search for all possible solutions in the given dimensions. The end of a recursion is reached when: number of Queens placed = size of rows and the positions of the queens in the current solution will be recorded in a list of solutions. Upon finishing, the number of solutions found will be displayed along with the first solution, additional display of other solutions may be requested by inputting the solution's index

Results and Discussion:

When it comes to using the brute force method there are two main things to account for; The root of the recursions only needs to start at each first column of the row (smaller dimension) and the number of solutions found for N-by-N boards matches the numbers officially agreed on. In regards to the knights move solution there are a few main problems faced when gathering solutions. The limitations are as follows the non-linear nature of the knights, the need to reposition to continue, if chained they can run into a dead end, and that it has to tabulate all possible solutions. The main question that must be addressed is the balance between exploration vs exploitation. In our current model we have too much exploitation and not enough exploration.

Conclusion:

In order to get the most complete data when it came to setting up the effective solutions to the Nqueens problem. That being said it was also a much slower way of getting the correct solutions. The other solution being the knights move solution which provided a much faster solution however left a lot of data out due to being limited by the way that the knight moves around on an infinitely increasing and decreasing board.