Braden Hoagland

Eight Queens Write-Up

Eight queens is a programming puzzle that challenges you to place eight queens on a chess board such that no two queens can attack each other. It is a specific case of the n queens problem, which involves placing n queens on an n x n board (“8 Queens”).

Brute force isn’t a viable option because we would have to loop through n! / (n - r)! = 64! / 56! = 178,462,987,637,760 layouts of queens on the board. The problem can be simplified by noticing that only one queen can be in a distinct row or column at a time. Because of this, we know that we only have to search through n! = 8! = 40,320 layouts (“Eight Queens”). These layouts have a queen in each row and column, and we have to search through them to find which ones don’t have any two queens on the same diagonal.

As we iterate through these layouts, we check to see if any queens are on the same diagonal. If none are, we save that layout to our set of solutions.

A picture containing object, checker

Description generated with very high confidence

My program was able to find all 92 solutions to the 8 queens problem using both non-recursive and recursive methods. I used this website (https://www.cs.usfca.edu/~galles/visualization/RecQueens.html) to see a visual play through of each iteration of a working recursive function. There is code on the left hand side of the website, but I didn’t look at it. After I had finished my algorithm, I went back to see theirs, and I realized that I could’ve made my code easier to read if I hadn’t made my moveToSafeSpot() helper method.

Works Cited

“Eight Queens.” *DataGenetics*, 2013, www.datagenetics.com/blog/august42012/.

“8 Queens.” *Brain Metrix*, 2016, www.brainmetrix.com/8-queens/.