Domino Tiling an N-Queens Chessboard

Christopher Sakaliyski

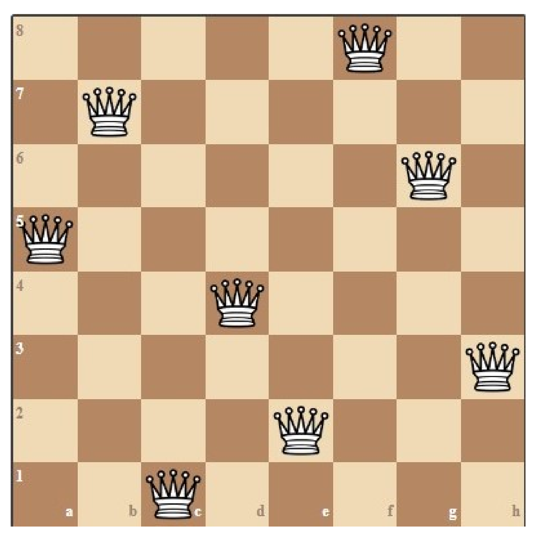
Whitney M. Young Magnet High School

QED 2022: Computer Science

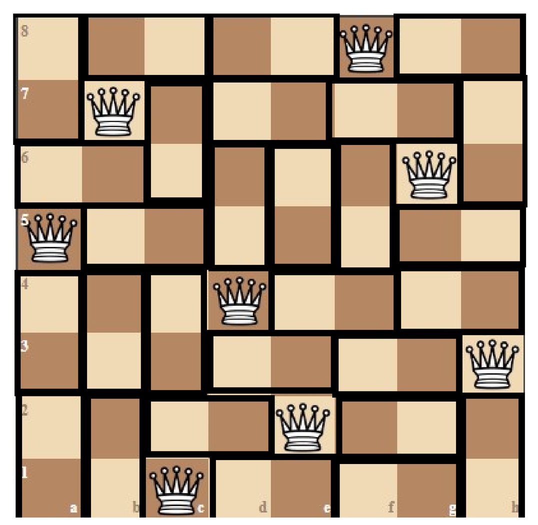
Introduction and Definition

N-Queens Problem

Place n queens on an n by n chessboard such that no two queens are attacking each other. The statement is equivalent to placing n queens on an n by n board such that they are all on different rows, different columns, and different diagonals. Although there are $\binom{64}{8} = 4,426,165,368$ ways to place 8 queens on an 8 by 8 chessboard without restrictions, only 92 of these are solutions to the problem. The number of fundamental solutions, or solutions that are not rotations or reflections of each other, has been shown to be only 12 for $n = 8$.



(a) No three queens in a line.



(b) Domino tiling of Figure 1a.

Figure 1: Example of a solution of 8-queens problem and one possible domino tiling.

Domino Tiling

The 1x2 domino tiling problem originates from Tatami, which are Japanese floor mats in the shape of a 1x2 rectangle. These mats are used to tile rooms, but with a bit more restriction on the tiling. On a n by n chessboard, 1x2 tiles are needed to be placed on the board such that they don't overlap and fill up the entire board. A form of this is mutilated tiling, in which some squares are removed from the board and the rest need to be tiled.

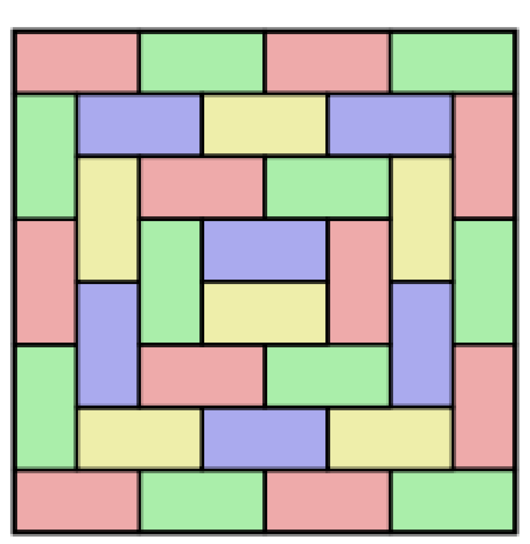


Figure 2: Tatami tiling of an 8x8 square.

Mutilated Chessboards

Mutilated chessboards are chessboards in which some squares are missing/mutilated. Since a chessboard is made up of alternating black and white squares, each 1x2 tile will cover one black square and one white square. The necessary condition for a mutilated chessboard to be tileable is that the number of black and white squares are equal. However, the mutilations can split the board into groups which are disjoint and not tileable. In the solutions to the n-queens puzzle, the queens (mutilated squares) are so far apart that the entire board is connected.

Algorithm

The process we use is backtracking. For every column, the program tries to place a queen in all squares which satisfy the constraints and covers that square. The program moves onto the next column and repeats. After the program places all queens, it moves onto tiling.

For every square that is not already covered, starting from the top left and going down row by row, the program tries to place a horizontal domino or a vertical domino on that square. After a domino is placed, the squares it is on are marked as covered. Once the entire board is filled, the counter of solutions is incremented by 1.

```
placeTiles(current x, current y)
{
    if all tiles have been placed
    {
        add one to counter of solutions
        break
    }
    if the current square has been covered
    {
        placeTiles(next x, next y)
    }
    set the tile to be covered in the boolean array
    if it is possible to place a left-right tile
    {
        place the left-right tile
        placeTiles(next x, next y)
        remove the tile
    }
    if it is possible to place an up-down tile
    {
        place the up-down tile
        placeTiles(next x, next y)
        remove the tile
    }
    set the current square to not be covered
}
placeQueens(current column)
{
    if current column is n
    {
        if number of black squares = number of white squares
        {
            placeTiles(0, 0)
        }
        break
    }
    for all squares in the current column
    {
        check if the square is attacked by any queens in the same row
        check if the square is attacked by any queens on both diagonals
        if the square is not attacked
        {
            place the queen
            mark the square as covered
            placeQueens(next column)
            remove the queen
            mark the square as not covered
        }
    }
}
```

Necessary Condition

Table 1: Exact number of white/black queens and squares when placing n queens on n by n board, to be 1x2 tileable.

n	All Squares	White Squares	Black Squares	White Queens	Black Queens
$4k$	$16k^2$	$8k^2$	$8k^2$	$2k$	$2k$
$4k+1$	$16k^2+8k+1$	$8k^2+4k+1$	$8k^2+4k$	$2k+1$	$2k$
$4k+2$	$16k^2+16k+4$	$8k^2+8k+2$	$8k^2+8k+2$	$2k+1$	$2k+1$
$4k+3$	$16k^2+24k+9$	$8k^2+12k+5$	$8k^2+12k+4$	$2k+2$	$2k+1$

Results

Table 2: Exact number of solutions for placing n queens on n by n board.

n	All solutions	Tileable Solutions
2	0	0
3	0	0
4	2	2
5	10	8
6	4	0
7	40	0
8	92	92
9	352	304
10	724	0
11	2,680	0
12	14,200	12,996
13	73,712	58,392
14	365,596	0
15	2,279,184	0

Table 3: Possible numbers of white and black queens along with the possible numbers of remaining white and black squares and arrangements on an 8 by 8 chessboard.

White Queens	Black Queens	White Squares	Black Squares	Arrangements
4	4	28	28	92

Table 4: Possible numbers of white and black queens along with the possible numbers of remaining white and black squares and arrangements on an 15 by 15 chessboard.

White Queens	Black Queens	White Squares	Black Squares	Arrangements
5	10	108	102	33,912
7	8	106	104	1,739,936
9	6	104	106	504,776
11	4	102	108	560

Conclusion

We devised and implemented efficient algorithms which compute all solutions to the n-queens problem and calculate the number of domino tilings for a given solution, all for $n \in [2, 15]$. Based on the numerical results our program has given us, we conjectured the following:

- An n-queens chessboard can be tiled if the number of white queens and black queens is equivalent. This is interesting as this condition is not sufficient for all mutilated chessboards.
- $n \equiv 0, 1 \pmod 4$: Some but not all arrangements of queens can be tiled.
- $n \equiv 2, 3 \pmod 4$: None of the arrangements of queens can be tiled.

Code and Data

The source code used to solve the two problems and obtain the results in the paper can be found at https://github.com/cks524/QED2022.