8 Queens Puzzle

Anna Clancy

December 18, 2014

1 The puzzle

The puzzle was first published by chess player Max Bezzel in 1848 and the first solution was completed in 1850 by Franz Nuack. [5] To complete the puzzle you have to place 8 queens on an 8×8 chessboard so that none of them can attack any of the others in only one move; so that none of them occupy the same row, column or diagonal line as in Figure 1. I intend to find a solution to this puzzle using sage. To represent the squares on the chess board, I have used (x, y) coordinates.

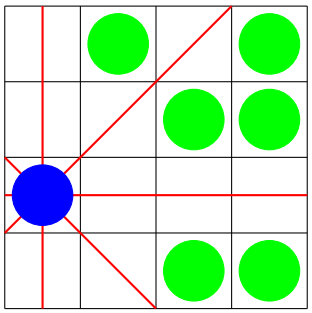

Figure 1: An existing queen (blue) and the possible positions for a new queen (green)

2 Conditions on placing new queens

For a new queen to be placed on the board:

- It must have a different x coordinate to all the existing queens
- It must have a different y coordinate to all the existing queens
- The difference of the x coordinates of the new queen and every existing one can't equal the difference of their y coordinates

The following code creates a function that returns True if it is possible to place a queen in any given square:

```
def possible(x, y):
    for [a, b] in solution:
        if y == b:
            return False
        if abs(x - a) == abs(y - b):
            return False
    return True
```

This piece of code is a slightly altered version of a piece of code from http://www.prasannatech.net/2012/07/eight-queens-python.html [3]

3 Finding a solution

A solution can be obtained using the following piece of code and the function above:

```
solution = []
tried = []


def solve(x, y=1):
    if x < 9 and y < 9:
        if possible(x, y) and [x, y] not in tried:
            solution.append([x, y])
        else:
            return solve(x + 1)
            return solve(x, y + 1)
    elif y > 8:
        for [c, d] in tried[:]:
            if solution[x - 2][0] < c:
                tried.remove([c, d])
        tried.append(solution[x - 2])
        solution.remove(solution[x - 2])
        return solve(x - 1)
    else:
        print solution

solve(1)
```

In order to save room I have left out some of the comments, the full code can be found here [1] This is the solution that it gives:

```
[[1, 1], [2, 5], [3, 8], [4, 6], [5, 3], [6, 7], [7, 2], [8, 4]]
```

Figure 2: The solution above depicted on a draughts board

3.1 Backtracking algorithm

This type of algorithm is known as a backtracking algorithm, which is also recursive. To find a solution, the code goes through each column placing a queen in the first possible square. It does this until it reaches a column where it can't place a new queen, at which point it goes back to the previous column and puts a queen in a the next available square and carries on. It repeats this process until a solution is found. By doing this it checks all the possible configurations for the queens if they are restricted to one per column and finding a solution hasn't become impossible. [4] This process is shown on a 4×4 board in Figure 3.

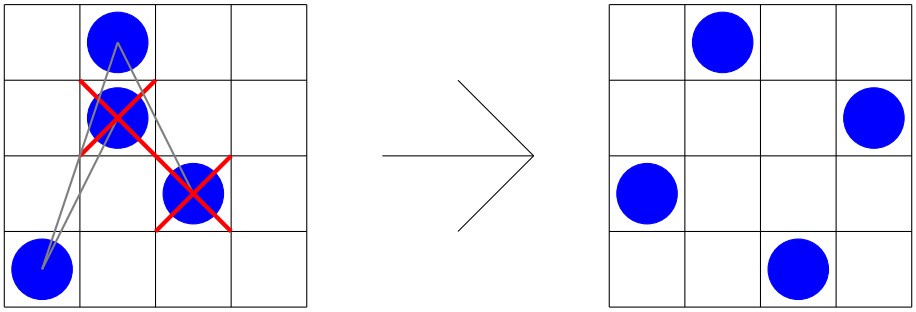

Figure 3: Backtracking on a 4×4 board

3.2 N-Queens version

Franz Nuack also generalised this puzzle to the n-queens version, where you have to place n queens on an $n \times n$ chessboard. [5] The above function can be modified to find a solution to a board of any size by adding another argument n, changing the 9s in the first line of the function to $n + 1$s, and changing the 8 in the elif statement to n.

4 Recursion limits

So far, no one has found a formula for the exact number of solutions to these puzzles. For the 8 queens version of the puzzle, there are $64C8 = 4426165368$ possible ways to arrange 8 queens on an 8×8 chessboard, although if you restrict it to one queen per column, as in the solve function used previously, it reduces this number to $8^8 = 16777216$, and there are only 92 solutions. The backtracking algorithm is more efficient computationally than one that would check every arrangement because it discards an arrangement when it can't place a new queen in the next column. However it does still use a lot of recursion: just finding the solution above exceeds the default recursion limit, so the following piece of code, which I found at http://stackoverflow.com/questions/3323001/maximum-recursion-depth [2], has to be included at the beginning to allow enough recursion for the solution to be found:

```
from sys import setrecursionlimit
setrecursionlimit(1500)
```

References

[1] Anna Clancy. Code for a solution to the eight queens puzzle.sagews.

[2] Stack Overflow. Python - maximum recursion depth? - stack overflow, 2010. [Online; accessed 18-December-2014].

[3] Prasanna Seshadri. An easy to understand solution to the eight queens problem in python, 2012. [Online; accessed 18-December-2014].

[4] Wikipedia. Backtracking - wikipedia the free encyclopedia, 2014. [Online; accessed 18-December-2014].

[5] Wikipedia. Eight queens puzzle - wikipedia the free encyclopedia, 2014. [Online; accessed 18-December-2014].