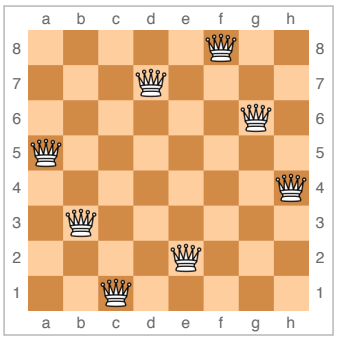
Eight queens puzzle

The **eight queens puzzle** is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other. Thus, a solution requires that no two queens share the same row, column, or diagonal. The eight queens puzzle is an example of the more general ***n* queens problem** of placing n non-attacking queens on an $n \times n$ chessboard, for which solutions exist for all natural numbers n with the exception of $n=2$ and $n=3$.[1]



The only symmetrical solution to the eight queens puzzle (except for rotations and reflections of itself).

Contents

History

Chess composer Max Bezzel published the eight queens puzzle in 1848. Franz Nauck published the first solutions in 1850.[2] Nauck also extended the puzzle to the n queens problem, with n queens on a chessboard of $n \times n$ squares.

Since then, many mathematicians, including Carl Friedrich Gauss, have worked on both the eight queens puzzle and its generalized n-queens version. In 1874, S. Gunther proposed a method using determinants to find solutions.[2] J.W.L. Glaisher refined Gunther's approach.

In 1972, Edsger Dijkstra used this problem to illustrate the power of what he called structured programming. He published a highly detailed description of a depth-first backtracking algorithm.[2]

Solution construction

The problem can be quite computationally expensive, as there are 4,426,165,368 (*i.e.*, ${}_{64}C_8$) possible arrangements of eight queens on an 8×8 board, but only 92 solutions. It is possible to use shortcuts that reduce computational requirements or rules of thumb that avoids brute-force computational techniques. For example, by applying a simple rule that constrains each queen to a single column (or row), though still considered brute force, it is possible to reduce the number of possibilities to 16,777,216 (that is, 8^8) possible combinations. Generating permutations further reduces the possibilities to just 40,320 (that is, 8!), which are then checked for diagonal attacks.

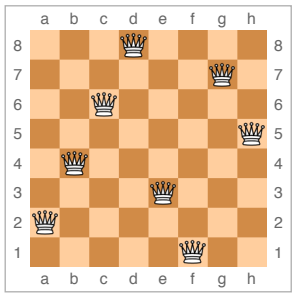
Martin Richards published a program to count solutions to the n-queens problem using bitwise operations.[3]. However, this solution has already been published by Zongyan Qiu[4].

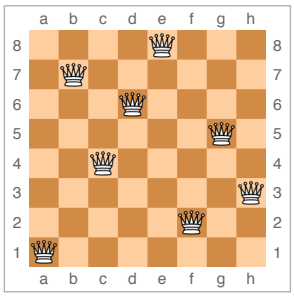
Solutions

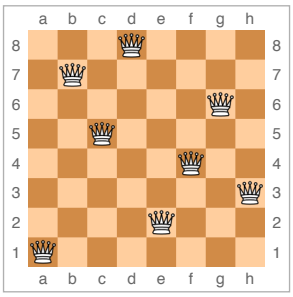
The eight queens puzzle has 92 distinct solutions. If solutions that differ only by the symmetry operations of rotation and reflection of the board are counted as one, the puzzle has 12 solutions. These are called *fundamental* solutions; representatives of each are shown below.

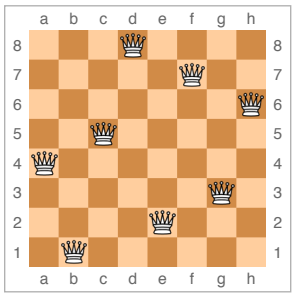
A fundamental solution usually has eight variants (including its original form) obtained by rotating 90, 180, or 270° and then reflecting each of the four rotational variants in a mirror in a fixed position. However, should a solution be equivalent to its own 90° rotation (as happens to one solution with five queens on a 5×5 board), that fundamental solution will have only two variants (itself and its reflection). Should a solution be equivalent to its own 180° rotation (but not to its 90° rotation), it will have four variants (itself and its reflection, its 90° rotation and the reflection of that). If $n > 1$, it is not possible for a solution to be equivalent to its own reflection because that would require two queens to be facing each other. Of the 12 fundamental solutions to the problem with eight queens on an 8×8 board, exactly one (solution 12 below) is equal to its own 180° rotation, and none is equal to its 90° rotation; thus, the number of distinct solutions is 11×8 + 1×4 = 92 (where the 8 is derived from four 90° rotational positions and their reflections, and the 4 is derived from two 180° rotational positions and their reflections).

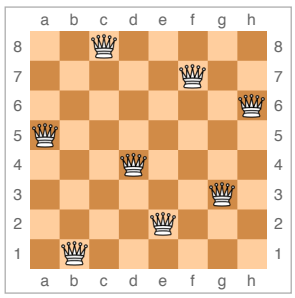
The different fundamental solutions are presented below:

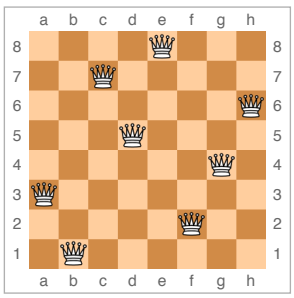

Solution 1

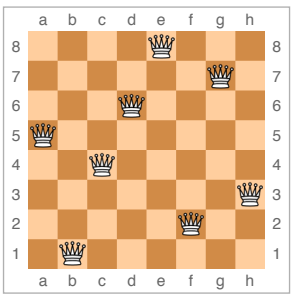

Solution 2

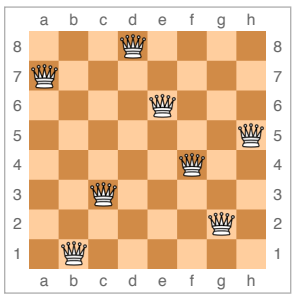

Solution 3

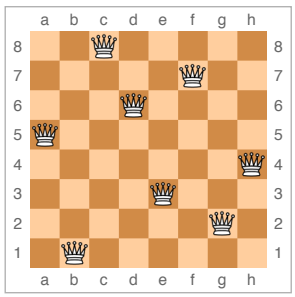

Solution 4

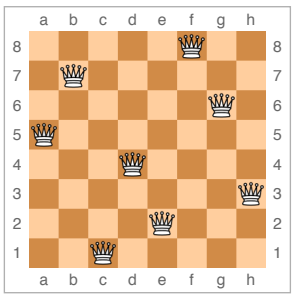

Solution 5

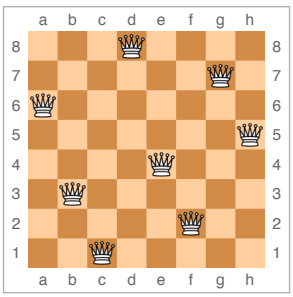

Solution 6

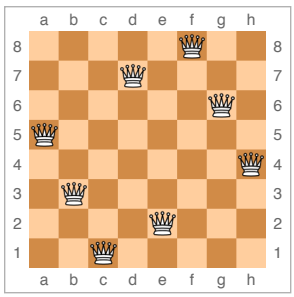

Solution 7


Solution 8


Solution 9


Solution 10

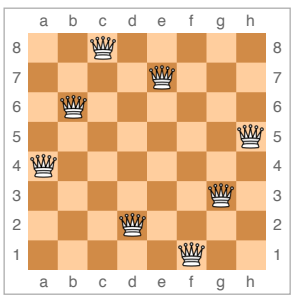

Solution 11

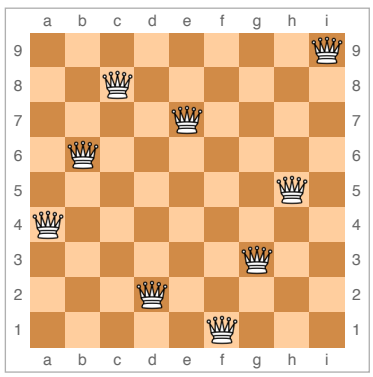

Solution 12

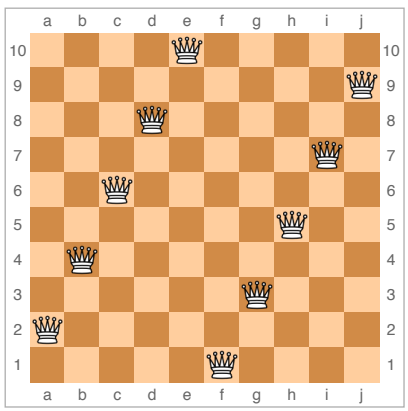
Solution 10 has the additional property that no three queens are in a straight line.

Explicit solutions

These brute-force algorithms to count the number of solutions are computationally manageable for n = 8, but would be intractable for problems of n ≥ 20, as 20! = 2.433 × 10^{18}. If the goal is to find a single solution then explicit solutions exist for all $n \geq 4$, requiring no combinatorial search whatsoever.[5] The explicit solutions exhibit stair-stepped patterns, as in the following examples for n = 8, 9 and 10:


Explicit solution for 8 queens


Explicit solution for 9 queens


Explicit solution for 10 queens

The examples above can be obtained with the following formulas. Let (i, j) be the square in column i and row j on the $n \times n$ chessboard, k an integer.

1. If n is even and $n \neq 6k + 2$, then place queens at $(i, 2i)$ and $(n/2 + i, 2i - 1)$ for $i = 1, 2, \ldots, n / 2$.
2. If n is even and $n \neq 6k$, then place queens at $(i, 1 + ((2i + n/2 - 3) \pmod n))$ and $(n + 1 - i, n - ((2i + n/2 - 3) \pmod n))$ for $i = 1, 2, \ldots, n / 2$.
3. If n is odd, then use one of the patterns above for $(n – 1)$ and add a queen at (n, n).

Another approach is

1. If the remainder from dividing n by 6 is not 2 or 3 then the list is simply all even numbers followed by all odd numbers not greater than N.
2. Otherwise, write separate lists of even and odd numbers (i. e. 2, 4, 6, 8 — 1, 3, 5, 7).
3. If the remainder is 2, swap 1 and 3 in odd list and move 5 to the end (i. e. **3, 1**, 7, **5**).

4. If the remainder is 3, move 2 to the end of even list and 1,3 to the end of odd list (i. e. 4, 6, 8, **2** — 5, 7, **1, 3**).
5. Append odd list to the even list and place queens in the rows given by these numbers, from left to right (i. e. a2, b4, c6, d8, e3, f1, g7, h5).

For $n = 8$ this results in fundamental solution 1 above. A few more examples follow.

- 14 queens (remainder 2): 2, 4, 6, 8, 10, 12, 14, 3, 1, 7, 9, 11, 13, 5.
- 15 queens (remainder 3): 4, 6, 8, 10, 12, 14, 2, 5, 7, 9, 11, 13, 15, 1, 3.
- 20 queens (remainder 2): 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 3, 1, 7, 9, 11, 13, 15, 17, 19, 5.

Counting solutions

The following table gives the number of solutions for placing n queens on an $n \times n$ board, both fundamental (sequence A002562 in the OEIS) and all (sequence A000170 in the OEIS), for n=1–10, 24–27.

***n*:**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**...**	**24**	**25**	**26**	**27**
fundamental:	1	0	0	1	2	1	6	12	46	92	...	28,439,272,956,934	275,986,683,743,434	2,789,712,466,510,289	29,363,791,967,678,199
all:	1	0	0	2	10	4	40	92	352	724	...	227,514,171,973,736	2,207,893,435,808,352	22,317,699,616,364,044	234,907,967,154,122,528

Note that the six queens puzzle has fewer solutions than the five queens puzzle.

There is currently no known formula for the exact number of solutions, or even for its asymptotic behaviour. Currently the 27x27 board is the highest-order board that has been completely enumerated.[6] Finding a single solution for a bigger board is not difficult.

Related problems

- Higher dimensions

  Find the number of non-attacking queens that can be placed in a d-dimensional chess space of size n. More than n queens can be placed in some higher dimensions (the smallest example is four non-attacking queens in a 3 × 3 × 3 chess space), and it is in fact known that for any k, there are higher dimensions where n^k queens do not suffice to attack all spaces.[7]

- Using pieces other than queens

  On an 8×8 board one can place 32 knights, or 14 bishops, 16 kings or eight rooks, so that no two pieces attack each other. Fairy chess pieces have also been substituted for queens. In the case of knights, an easy solution is to place one on each square of a given color, since they move only to the opposite color. The solution is also easy for rooks and kings. Eight rooks can be placed along a long diagonal (amongst thousands of other solutions), and 16 kings are placed on the board by dividing it into 2 by 2 squares and placing the kings at equivalent points on each square.

- Permutation matrix

  In mathematics, a permutation matrix can be regarded geometrically as a set of n points lying on the squares of a nxn chessboard, such that each row or column contains only one point. Thus, an order-n permutation matrix is a solution to an n-rooks puzzle.

- Nonstandard boards

  Pólya studied the n queens problem on a toroidal ("donut-shaped") board and showed that there is a solution on an n×n board if and only if n is not divisible by 2 or 3.[8] In 2009 Pearson and Pearson algorithmically populated three-dimensional boards (http://queens.lyndenlea.info/beyond2d.php) *(n×n×n)* with n^2 queens, and proposed that multiples of these can yield solutions for a four-dimensional version of the puzzle.

- Domination

  Given an n×n board, the **domination number** is the minimum number of queens (or other pieces) needed to attack or occupy every square. For n=8 the queen's domination number is 5.

- Nine queens problem (http://www.chessvariants.org/problems.dir/9queens.html)

  Place nine queens and one pawn on an 8×8 board in such a way that queens don't attack each other. Further generalization of the problem (complete solution is currently unknown): given an n×n chess board and $m > n$ queens, find the minimum number of pawns, so that the m queens and the pawns can be set up on the board in such a way that no two queens attack each other.

- Queens and knights problem (http://www.vector.org.uk/archive/v213/hui213.htm)

  Place m queens and m knights on an n×n board so that no piece attacks another.

- Magic squares

  In 1992, Demirörs, Rafraf, and Tanik published a method for converting some magic squares into n-queens solutions, and vice versa.[9]

- Latin squares

  In an n×n matrix, place each digit 1 through n in n locations in the matrix so that no two instances of the same digit are in the same row or column.

- Exact cover

  Consider a matrix with one primary column for each of the n ranks of the board, one primary column for each of the n files, and one secondary column for each of the $4n - 6$ nontrivial diagonals of the board. The matrix has n^2 rows: one for each possible queen placement, and each row has a 1 in the columns corresponding to that square's rank, file, and diagonals and a 0 in all the other columns. Then the n queens problem is equivalent to choosing a subset of the rows of this matrix such that every primary column has a 1 in precisely one of the chosen rows and every secondary column has a 1 in at most one of the chosen rows; this is an example of a generalized exact cover problem, of which sudoku is another example.

- n-Queens Completion

A 2017 paper[10] investigated the problem “Given an $n \times n$ chessboard on which some queens are already placed, can you place a queen in every remaining row so that no two queens attack each other?” and several related ones. The authors asserted these problems are NP-complete, so related to a million dollar Millennium Prize[11], and #P-complete.

Exercise in algorithm design

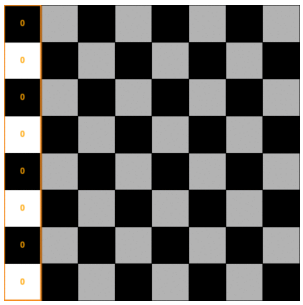
Finding all solutions to the eight queens puzzle is a good example of a simple but nontrivial problem. For this reason, it is often used as an example problem for various programming techniques, including nontraditional approaches such as constraint programming, logic programming or genetic algorithms. Most often, it is used as an example of a problem that can be solved with a recursive algorithm, by phrasing the n queens problem inductively in terms of adding a single queen to any solution to the problem of placing $n-1$ queens on an n-by-n chessboard. The induction bottoms out with the solution to the 'problem' of placing 0 queens on the chessboard, which is the empty chessboard.

This technique is much more efficient than the naïve brute-force search algorithm, which considers all $64^8 = 2^{48} = 281{,}474{,}976{,}710{,}656$ possible blind placements of eight queens, and then filters these to remove all placements that place two queens either on the same square (leaving only $64!/56! = 178{,}462{,}987{,}637{,}760$ possible placements) or in mutually attacking positions. This very poor algorithm will, among other things, produce the same results over and over again in all the different permutations of the assignments of the eight queens, as well as repeating the same computations over and over again for the different sub-sets of each solution. A better brute-force algorithm places a single queen on each row, leading to only $8^8 = 2^{24} = 16{,}777{,}216$ blind placements.

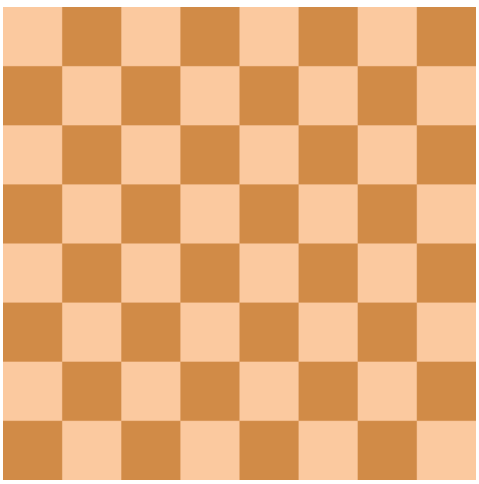
It is possible to do much better than this. One algorithm solves the eight rooks puzzle by generating the permutations of the numbers 1 through 8 (of which there are 8! = 40,320), and uses the elements of each permutation as indices to place a queen on each row. Then it rejects those boards with diagonal attacking positions. The backtracking depth-first search program, a slight improvement on the permutation method, constructs the search tree by considering one row of the board at a time, eliminating most nonsolution board positions at a very early stage in their construction. Because it rejects rook and diagonal attacks even on incomplete boards, it examines only 15,720 possible queen placements. A further improvement, which examines only 5,508 possible queen placements, is to combine the permutation based method with the early pruning method: the permutations are generated depth-first, and the search space is pruned if the partial permutation produces a diagonal attack. Constraint programming can also be very effective on this problem.

An alternative to exhaustive search is an 'iterative repair' algorithm, which typically starts with all queens on the board, for example with one queen per column.[12] It then counts the number of conflicts (attacks), and uses a heuristic to determine how to improve the placement of the queens. The 'minimum-conflicts' heuristic — moving the piece with the largest number of conflicts to the square in the same column where the number of conflicts is smallest — is particularly effective: it finds a solution to the 1,000,000 queen problem in less than 50 steps on average. This assumes that the initial configuration is 'reasonably good' — if a million queens all start in the same row, it will obviously take at least 999,999 steps to fix it. A 'reasonably good' starting point can for instance be found by putting each queen in its own row and column so that it conflicts with the smallest number of queens already on the board.

Note that 'iterative repair', unlike the 'backtracking' search outlined above, does not guarantee a solution: like all hillclimbing (i.e., greedy) procedures, it may get stuck on a local optimum (in which case the algorithm may be restarted with a different initial configuration). On the other hand, it can solve problem sizes that are several orders of magnitude beyond the scope of a depth-first search.



min-conflicts solution to 8 queens



This animation illustrates backtracking to solve the problem. A queen is placed in a column that is known not to cause conflict. If a column is not found the program returns to the last good state and then tries a different column.

Sample program

The following is a Pascal program by Niklaus Wirth in 1976.[13] It finds one solution to the eight queens problem.

```
program eightqueen1(output);

var i : integer; q : boolean;
    a : array[ 1 .. 8] of boolean;
    b : array[ 2 .. 16] of boolean;
    c : array[ -7 .. 7] of boolean;
    x : array[ 1 .. 8] of integer;

procedure try( i : integer; var q : boolean);
    var j : integer;
    begin
    j := 0;
    repeat
        j := j + 1;
        q := false;
        if a[ j] and b[ i + j] and c[ i - j] then
            begin
            x[ i    ] := j;
            a[ j    ] := false;
            b[ i + j] := false;
            c[ i - j] := false;
            if i < 8 then
                begin
                try( i + 1, q);
```

```
                if not q then
                    begin
                    a[ j] := true;
                    b[ i + j] := true;
                    c[ i - j] := true;
                    end
                end
            else
                q := true
            end
    until q or (j = 8);
    end;

begin
for i :=  1 to  8 do a[ i] := true;
for i :=  2 to 16 do b[ i] := true;
for i := -7 to  7 do c[ i] := true;
try( 1, q);
if q then
    for i := 1 to 8 do write( x[ i]:4);
writeln
end.
```

See also

- Mathematical game
- Mathematical puzzle
- No-three-in-line problem
- Rook polynomial

References

1. E. J. Hoffman et al., "Construction for the Solutions of the m Queens Problem". *Mathematics Magazine*, Vol. XX (1969), pp. 66–72. [1] (http://penguin.ewu.edu/~trolfe/QueenLasVegas/Hoffman.pdf)
2. W. W. Rouse Ball (1960) "The Eight Queens Problem", in *Mathematical Recreations and Essays*, Macmillan, New York, pp. 165–171.
3. Martin Richards. Backtracking Algorithms in MCPL using Bit Patterns and Recursion. University of Cambridge Computer Laboratory. http://www.cl.cam.ac.uk/~mr10/backtrk.pdf
4. Qiu, Zongyan (February 2002). "Bit-vector encoding of n-queen problem". *ACM SIGPLAN Notices*. **37** (2): 68-70. doi:10.1145/568600.568613 (https://doi.org/10.1145%2F568600.568613).
5. *Explicit Solutions to the N-Queens Problem for all N* (http://delivery.acm.org/10.1145/130000/122322/p7-bernhardsson.pdf?ip=128.125.133.201&CFID=44352328&CFTOKEN=62267842&__acm__=1316815405_9a82b00e821cdd41080729f6f3f40975), Bo Bernhardsson (1991), Department of Automatic Control, Lund Institute of Technology, Sweden.
6. The Q27 Project (http://github.com/preusser/q27)
7. J. Barr and S. Rao (2006), The n-Queens Problem in Higher Dimensions, Elemente der Mathematik, vol 61 (4), pp. 133–137.
8. G. Pólya, Uber die "doppelt-periodischen" Losungen des n-Damen-Problems, George Pólya: Collected papers Vol. IV, G-C. Rota, ed., MIT Press, Cambridge, London, 1984, pp. 237–247
9. O. Demirörs, N. Rafraf, and M.M. Tanik. Obtaining n-queens solutions from magic squares and constructing magic squares from n-queens solutions. Journal of Recreational Mathematics, 24:272–280, 1992
10. Gent, Ian P.; Jefferson, Christopher; Nightingale, Peter (August 2017). "Complexity of *n*-Queens Completion" (http://jair.org/papers/paper5512.html). *Journal of Artificial Intelligence Research*. AAAI Press. **59**: 815–848. doi:10.1613/jair.5512 (https://doi.org/10.1613%2Fjair.5512). ISSN 1076-9757 (https://www.worldcat.org/issn/1076-9757). Retrieved September 7, 2017.
11. "The 8-Queens Puzzle" (http://www.claymath.org/events/news/8-queens-puzzle). *www.claymath.org*. Clay Mathematics Institute. September 2, 2017. Retrieved September 7, 2017.
12. A Polynomial Time Algorithm for the N-Queen Problem (http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4DC9292839FE7B1AFABA1EDB8183242C?doi=10.1.1.57.4685&rep=rep1&type=pdf) by Rok Sosic and Jun Gu, 1990. Describes run time for up to 500,000 Queens which was the max they could run due to memory constraints.
13. Wirth, 1976, p. 145

Further reading

- Bell, Jordan; Stevens, Brett (2009). "A survey of known results and research areas for **n**-queens". *Discrete Mathematics*. **309** (1): 1–31. doi:10.1016/j.disc.2007.12.043 (https://doi.org/10.1016%2Fj.disc.2007.12.043).
- Watkins, John J. (2004). *Across the Board: The Mathematics of Chess Problems*. Princeton: Princeton University Press. ISBN 0-691-11503-6.
- O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare *Structured Programming*, Academic Press, London, 1972 ISBN 0-12-200550-3 see pp 72–82 for Dijkstra's solution of the 8 Queens problem.
- Allison, L.; Yee, C.N.; McGaughey, M. (1988). "Three Dimensional NxN-Queens Problems" (http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Recn/Queens3D/). Department of Computer Science, Monash University, Australia.
- Nudelman, S. (1995). "The Modular N-Queens Problem in Higher Dimensions". *Discrete Mathematics*. **146** (1-3): 159–167. doi:10.1016/0012-365X(94)00161-5 (https://doi.org/10.1016%2F0012-365X%2894%2900161-5).
- Engelhardt, M. (August 2010). "Der Stammbaum der Lösungen des Damenproblems (in German, means The pedigree chart of solutions to the 8-queens problem" (http://www.spektrum.de/artikel/1037434&_z=798888). *Spektrum der Wissenschaft*: 68–71.
- *On The Modular N-Queen Problem in Higher Dimensions* (http://www.liacs.nl/~kosters/nqueens/papers/gomez2004.pdf), Ricardo Gomez, Juan Jose Montellano and Ricardo Strausz (2004), Instituto de Matematicas, Area de la Investigacion Cientifica, Circuito Exterior, Ciudad Universitaria, Mexico.
- Wirth, Niklaus (1976), *Algorithms + Data Structures = Programs*, Prentice-Hall, ISBN 0-13-022418-9

External links

- Weisstein, Eric W. "Queens Problem" (http://mathworld.wolfram.com/QueensProblem.html). *MathWorld*.
- queens-cpm (https://github.com/sblendorio/queens-cpm) on GitHub Eight Queens Puzzle in Turbo Pascal for CP/M