Beyond the 8 Queens Problem

Dr. Yury Zavarovsky

The 8 Queens Problem is a well-known problem that asks you to place eight chess queens on an 8×8 chessboard so that no two queens can attack each other. In this application, we consider the more general version of placing m chess queens on an $n \times n$ chessboard. The problem has a solution if $n > 3$ and $m \leq n$.

$$m \leq n \tag{1}$$

The work consists of two procedures. The first procedure , Queens, takes m (the number of queens) and n (the size of the chess board), returns the total number of solutions, and saves a complete list of all solutions (global variable S).

The second procedure, QueensPic, displays selected solutions on a chessboard. Calls are all of the form QueensPic(M, t), where M is a list of one or more solutions from S, and t is the number of solutions in each row of the display.

```
Queens := proc (m::posint, n::posint)
local It, K, l, L, M, P; global S, p, q;
     It := proc (L)
          local P, k, i, j;
          M := []; k := nops(L[1]);
          for i in L do
               for j to n do
                    if convert([seq(j <> i[s, 2], s = 1 .. k)], `and`) and
                         convert([seq(l[k+1]-i[s, 1] <> i[s, 2]-j, s = 1 .. k)], `and`) and convert([seq(l[k+1]-i[s, 1] <> j-i[s, 2], s = 1 .. k)], `and`) then
                         M := [op(M), [op(i), [l[k+1], j]]]
                    end if
               end do
          end do;
          M
     end proc;
     K := combinat:-choose([`$`(1 .. n)], m); S := [];
     for l in K do
          P := []; L := [seq([[l[1], i]], i = 1 .. n)];
          P := [op(P), op((It@@(m-1))(L))];
          S := [op(S), op(P)]
     end do;
     p := args[1]; q := args[2];
     nops(S)
end proc:

QueensPic := proc (M, t::posint)
local m, n, HL, VL, T, A, N;
uses plottools, plots;
     m := p;
     n := q;
     N := nops(args[1]);
     HL := seq(plottools:-line([.5, .5+k], [.5+n, .5+k], color = black, thickness = 2), k = 0 .. n);
     VL := seq(plottools:-line([.5+k, .5], [.5+k, .5+n], color = black, thickness = 2), k = 0 .. n);
     T := [seq(plots:-textplot([seq([op(M[i, j]), Q], j = 1 .. m)], color = red, font = [TIMES, ROMAN, 24]), i = 1 .. N)];
     if m <= n and 3 < n then
          A := seq(plots:-display(HL, VL, T[k], axes = none, scaling = constrained), k = 1 .. N),
               seq(plots:-display(plot([[0, 0]]), axes = none, scaling = constrained), k = 1 .. t*ceil(N/t)-N);
          Matrix(ceil(N/t), t, [A]);
          plots:-display(%)
     end if
end proc:
```

Examples

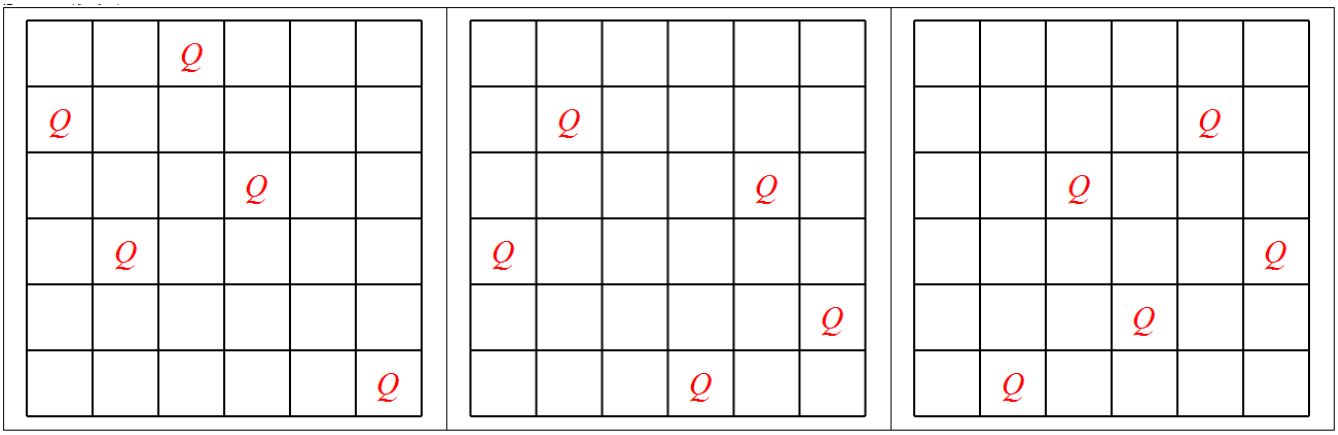
Queens(5, 6);

248 **(2)**

S[70], S[140], S[210];

[[1, 5], [2, 3], [3, 6], [4, 4], [6, 1]], [[1, 3], [2, 5], [4, 1], [5, 4], [6, 2]], [[2, 1], [3, 4], [4, 2], [5, 5], [6, 3]] **(3)**

QueensPic([%], 3);



Queens(8, 8);

92 **(4)**

S[64..65];

[[[1, 5], [2, 8], [3, 4], [4, 1], [5, 7], [6, 2], [7, 6], [8, 3]], [[1, 6], [2, 1], [3, 5], [4, 2], [5, 8], [6, 3], [7, 7], [8, 4]]] **(5)**

QueensPic(%, 2);