

Place n queens on an $n \times n$ chess board so that no two queens are in the same row, column, or diagonal.

Establish the problem.

- specifications – task, input, output, legal solution, optimal solution

Task:

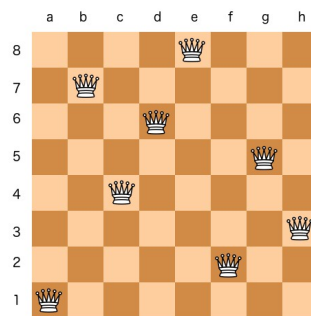
Place n queens on an $n \times n$ chess board so that no two queens are in the same row, column, or diagonal.

Input: n (number of queens to place)

Output: positions of the n queens

Legal solution: no two queens are in the same row, col, diag

- examples



Identify avenues of attack.

- known targets

Inherently exponential – $O(b^n)$

- approach

Series of choices.

- paradigms and patterns

Paradigm: backtracking.

Flavor: n/a

Pattern: process input – for each queen, put it on the board (figure out which row to put it in for the current column)

[produce output – for each position on the board, is there a queen there?]

Structural variation: find a legal solution

- the series of choices

which row the current queen goes in

Define the algorithm.

- size

original problem: n = number of queens, size of the board

rest of the problem (subproblem): number of queens left to place

number of choices: number of queens left

- generalize / define subproblems
 - partial solution

placements (which rows) for the first k queens (in the first k columns)

- alternatives

rows in the current column that aren't attacking an already-placed queen

- subproblem

task: placing the remaining $n-k$ queens on the $n \times n$ board (legally)

input: how many queens left to place, current column, placements (rows) of the first k queens

output: positions of all the queens (complete solution)

- base case(s)

size 0 – no choices left ($n-k=0$), all queens have been placed

return the partial (now complete) solution we were given

- main case

// job of the main case – make the next decision (try all the legal alternatives)

// for each legal alternative, make the choice, have a friend solve the rest of the problem, if they are successful → we have solution

for each row $1..n$

if a queen in the current column and that row doesn't attack an existing queen

add that queen to the partial solution in that position

have friend solve the rest

if they find a solution, return it

otherwise, remove the queen from the partial solution

(no solution has been found) return no solution

- top level
 - initial subproblem
 - setup
 - wrapup
- special cases
- algorithm

Show termination and correctness.

- termination

- making progress
- the end is reached
- correctness
 - establish the base case(s)
 - show the main case
 - final answer

Determine efficiency.

- implementation
- time and space
- room for improvement