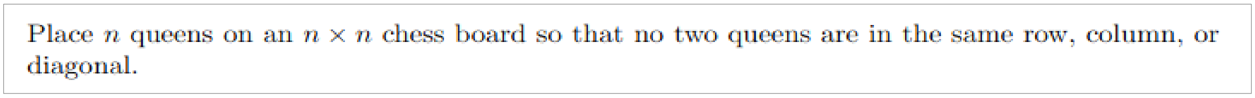
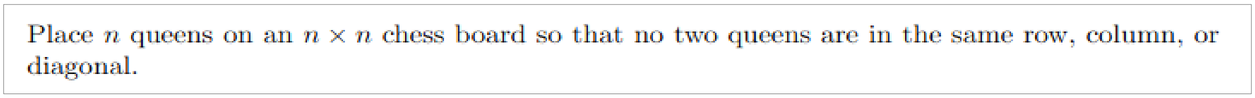


Place n queens on an $n \times n$ chess board so that no two queens are in the same row, column, or diagonal.

Establish the problem.

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

Task:



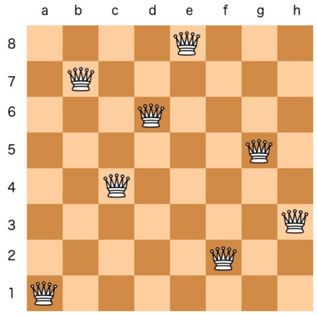
Place n queens on an $n \times n$ chess board so that no two queens are in the same row, column, or diagonal.

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.

- targets
- approach

Series of choices.

- paradigms and patterns

Paradigm: backtracking.

Flavor: n/a

Pattern: process input – for each queen, decide on its position

produce output – repeatedly find the next position for a queen

- the series of choices

deciding on positions, one queen at a time

~~height of tree – *n* because one choice per queen~~

~~branching factor – n^2 because n^2 places on the board~~

deciding on positions, one column at a time

for each column, decide on where that queen goes height of tree – *n* (n queens, n columns)

branching factor – *n* (rows per column)

Define the algorithm.

- size

n – number of queens

- generalize / define subproblems
    - partial solution

placement of the first *k* queens (in the first *k* columns)

-
    - alternatives

the *n* rows in the current column – legal if queens 1..k don't attack

-
    - subproblem

task: legally place the remaining queens given placement of first *k-1*

input: partial solution (placement of first *k-1* queens), the current queen *k*

output: positions of all *n* queens

legal:

```
nqueens ( placement of first k-1 queens, k )
  …
```

- base case(s)

- main case

```
for each row r 1..n
   place a queen in row r in column k
   result ← nqueens ( placement of k queens, k+1 )
   if result is a solution, return result
   remove queen from column k
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