1. specifications

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

input: n queens

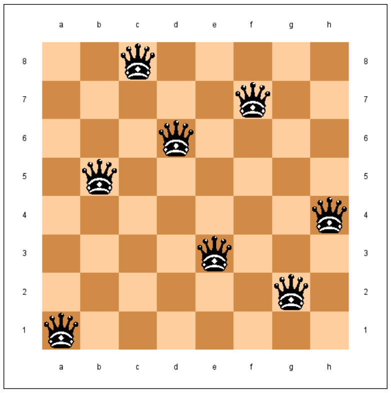
output: placement of queens on the board

legal solution: queens don't attack

2. size

3. examples

A solution for n=8:



4. targets

5. tactics

6. approaches

process input – for each queen, place it on the board

~~produce output – repeatedly find the next pos that has a queen~~

7. generalize / define subproblems
   a) partial solution – placement of some queens
   b) alternatives – all the empty squares on the board where the queen doesn't attack anything already there
   c) subproblem -

Place n-*k* queens on an *n*×*n* chess board so that no two queens are in the same row, column, or diagonal.

input: k queens already placed (and their positions), n-k remaining

output: placement of queens on the board legal solution: queens don’t attack

8. base case(s)

complete solution – n-k = 0 remaining to place or n already placed

9. main case

```
for each position on the board that doesn’t attack an existing queen
  place the next queen there
  nqueens(placement of k+1 queens,n-k-1 left)
```

10. top level
    a) initial subproblem
    b) setup
    c) wrapup

11. special cases

12. algorithm

13. termination
    a) making progress

    b) reaching the end

14. correctness
    a) establish the base case(s)

    b) show the main case

    c) final answer

15. implementation

16. time and space