

8.1 Queens on a chessboard

Find a set-up of 8 queens on a ~~FKHMERDUG~~ so that no queen threatens another one.

Ideas for generalizations:

- How does your solution “scale”? That is, if I ask you to solve n queens on a $n \times n$ board, will it be possible to do with your current solution or will you need to write a new one? (consider writing a solution for which this is very easy).
- How long does your program take as a function of n ? Is it linear $t \propto n$? Polynomial, $t \propto n^\alpha$ (if so, what is α ?), or exponential, $t = e^{an}$ (if so, what is a)? These are important questions to ask when you are coding a program that needs to work on large data.
- Can you think of ways to make your code faster? Use less memory?
- Find how many different solutions (arrangements of queens that is) are there. You will probably have to change you code to find this out. Can you also count without including rotations and mirror symmetries?
- How many placements of n queens on a $m \times m$ board are there? Can you find a nice formula? How about a few of the first examples? How about when $m = n$?
- What about other pieces? How many (truly different, or not) ways are there to put knights on the board?

MIT OpenCourseWare
<http://ocw.mit.edu>

18.S997 Introduction To MATLAB Programming
Fall 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.