

CSci 169 Programming HW 2

10 points

Python 1

Due Wed. Apr. 15 at the beginning of class

Submission instructions at the end

1. (10 points) Translate the C++ program here:

<http://math.scu.edu/~linnell/169resources/main.cpp> to Python (Quicksort):

Submission instructions: You will submit all of your .py files as email attachments, to

cs169scu@gmail.com (NOT Dr. Linnell's email!!)

This email should be sent before the beginning of class. Please name your files problem1.py, problem2.py, etc.

The subject line of the email should be "CS169 HW2 YourLastName YourIDNumber "

Note: The following problem will be due with Programming HW 3 due Wed 4/22, but I HIGHLY suggest you start it early (But wait to submit it with HW3). For many of you, this will be the hardest programming problem you've ever been asked to solve.

1. (15 points) In this problem, you will write a Python program to solve the 8-Queens problem. In chess, a queen can move any number of squares horizontally, vertically, or diagonally. The 8-queens problem is the problem of trying to place eight queens on an empty chessboard in such a way that no queen can attack any other queen. This problem is intriguing because there is no efficient algorithm known for solving the general problem. Rather, the straightforward algorithm of trying all possible placements is most often used in practice, with the only optimization being that each queen must be placed in a separate row and column:
 - a. Starting with the first row, try to place a queen in the current column.
 - b. If you can safely place a queen in that column, move on to the next column
 - c. If you are unable to safely place a queen in that column, go back to the previous column, and move that queen down to the next row where it can safely be placed. Move on to the next column.

Write a program in python to solve the 8-queens problem. Your program should produce as output an 8X8 diagram of the chessboard, with a 1 indicating the presence of a queen in a square and a 0 indicating the absence of a queen.

Hints: You can represent the board as either an 8X8 list or as a one-dimensional list with the *i*th item representing the row number of the queen in column *i*. You can solve this problem recursively or iteratively, but the recursive solution is usually much easier.