Homework 3

This homework is meant to give you feel for how different algorithms can affect runtime.

For this homework you will be required to implement two different sorting algorithms. You can choose from the ones we covered in class (not random sort) or use your own (there are lots if you spend some time searching online).

The only constraint on the two that you pick is that they must be in different complexity classes. Most likely you will need to find something that is $O(n^2)$ and $O(n\log n)$ but feel free to find something exotic or make up your own. *You must implement the sorting algorithms yourself.*

Once you have verified that your sorts are working properly (using tests), you will need to run a simulation and graph the results. Specifically, produce a graph with the following characteristics:

- The vertical axis is some measure of time
- The horizontal axis is N (size of set to sort)
- You have one line for each sort algorithm, showing how time goes up with N
- Everything is labeled appropriately

Try to pick an N such that the effect is visually noticeable. It should not take a very large increase to make this possible.

Bonus: Also graph quicksort. Note whether you are graphing average, best or worst case run-time. To test average run times try generating an array full of random numbers and sorting it. Do this a number of times and take the mean run-time.