

Implement BubbleSort and QuickSort for integers

- ◆ Instrument your code
 - Count number of assignments
 - Count number of conditionals

Test the time complexity of your algorithms as follows

- ◆ For sizes of 100, 200, 300, ... 1000
- ◆ Generate 100 random arrays
- ◆ Sort them using your code

You can use any language, e.g.: C, Python, Fortran, Lisp, Perl

Using the count data generated, illustrate the following:

- ◆ BubbleSort is $O(n^2)$ on average
- ◆ QuickSort is $O(n \log(n))$ on average

Consider the protein folding problem

- ◆ On average, how many bonds rotate in a non-terminal amino acid within a protein?
- ◆ Given that the main-chain dihedrals dominate the folding questions, for a protein of N amino acids, what is the size of the approximate configurational space?
- ◆ There are only about 50,000 proteins in the eukaryotic proteome even though the theoretical number of different peptides of length 10 is much larger.
- ◆ Given that typical proteins are often a few hundred amino acids, what are the implications of these numbers for protein folding algorithms?

What to turn in: a **single** PDF

(name your file sensibly: JaneSmith_BMI203_HW1.pdf)

- ◆ Readable listing of your code
- ◆ Input and output of both procedures on one example of size 100
- ◆ Graphical depiction of counts for assignments and conditionals for both functions
- ◆ Convincing argument (graphs are helpful!) that the algorithms' average case performance is as expected
- ◆ Answers and discussion of the protein folding questions

This assignment is a bit of a “gimme” so style counts!

Email enclosure to: ajain@jainlab.org