ICS-211 Lab

Assignment 9

Last assignment - yay!

Overview

- Very similar to assignment 1, but implementing merge, quick and heap sorts
  - Add new sort methods to the ArraySort class (there should be 6 in total)
  - Reuse a lot of test code
- All sort methods are “generic” (i.e., they should be able to sort Integers, Strings, Whatevers, …)
- Grading
  - 25% per sort method (75% total)
  - 25% for write-up/analysis

Analysis/Write-up

- Empirical data
  - Run and time sorts
  - You need to have implemented all six sorting methods
  - Organize and label your data (see example table below)
- Theoretical performance
  - How fast should each sort run in theory?
  - Best-case, worst-case?
- Theory vs. Reality
  - Does your measured data match the theoretical expectation?
  - If it doesn't what explanation could there be?
  - Questions worth answering:
    - What was the fastest sort?
    - Which sort within the same performance class was fastest?
    - Etc...

words/sort	**bubble**	**insertion**	**selection**	**merge**	**quick**	**heap**
1000	*time*					
5000						
10000						

Starter Code & Testing

- You *should* be able to use a lot of the code you wrote for A01
  - Same basic code for unit testing
  - Same basic code for timing the sort methods
- Provided code:
  - Test just uses a bubble sort; obviously you need to add tests for the new sorts
  - See the “a9” directory in my GitHub repo (https://github.com/psoulier/ics211-fall16)
  - Code to load words from text file (must be able to find the text files containing words)
  - Method to check if array is sorted (virtually the same as A01)
  - Simple unit test that shows how to load words and sort them
  - A `String` comparator
- Test your code
  - The code I have provided does not comprehensively test sort methods
  - However, if your code doesn’t pass the unit test I did provide, that’s not good