Jamie Grasley
ESOF-3251
#1164621
Lab II - Algorithms

General Info & Specifications

Compiled and built on MacOS using the OpenJDK version 23.0.1 from Oracle

Commands Used

For easy running, navigate to the folder with the two jar files in a terminal and run

1. java -jar TestDriver.jar
2. java -jar SortAnalyzer.jar

All other compilation has been done in an IDE for easy management

Files Included

- ReadMe file
    - ESOF_3251_LAB2_Part1_ReadMe_Jamie_Grasley_1164621_28_NOV_24.txt
- Folder with all classes
    - ESOF_3251_Jamie_Grasley_1164621_28_NOV_24.sort
- Jar Files for TestDriver and SortAnalyzer
    - ESOF_3251_LAB2_Part1_JAR_Jamie_Grasley_1164621_28_NOV_24 (folder)
        - TestDriver.jar
        - SortAnalyzer.jar
- Raw Java files folder
    - ESOF_3251_LAB2_Part1_Java_Jamie_Grasley_1164621_28_NOV_24

Challenges

Challenges initially included difficulties with debugging due to typos.

I also misunderstood the output from the TestDriver at first and thought I was doing something wrong but once I realized I was doing it correctly it worked fine.

I briefly had a data loss issue as well when dividing since I was dividing a long by an int. To solve this I casted the long to a double and then it worked fine.

Output

```
Last login: Thu Nov 28 23:36:15 on ttys000
jamiegrasley@Jamies-MacBook-Air-8 Project II  % java -jar TestDriver.jar
TESTING: insertion sort with sorted data
checking compares
                PASSED
TESTING: insertion sort with reverse data
checking compares
                PASSED
TESTING: insertion sort with random data
checking compares
                PASSED

TESTING: selection sort with sorted data
checking compares
                PASSED
TESTING: selection sort with reverse data
checking compares
                PASSED
TESTING: selection sort with random data
checking compares
                PASSED

TESTING: bubble sort with sorted data
checking compares
                PASSED
TESTING: bubble sort with reverse data
checking compares
                PASSED
TESTING: bubble sort with random data
checking compares
                PASSED

TESTING: quick sort with sorted data
checking compares
                PASSED
TESTING: quick sort with reverse data
checking compares
                PASSED
TESTING: quick sort with random data
checking compares
                PASSED

Testing Analyzer
                PASSED
jamiegrasley@Jamies-MacBook-Air-8 Project II  %
```

TestDriver shows that all tests pass (the expect vs actual only appears if anything is incorrect)

```
jamiegrasley@Jamies-MacBook-Air-8 Project II  % java -jar SortAnalyzer.jar
INSERTION sort with SORTED data
Sizes:  100 200 400 800 1600
Compares:  99 199 399 799 1599
Ratios:  0.9900 0.9950 0.9975 0.9988 0.9994 O(N) error 0.0029

INSERTION sort with REVERSE data
Sizes:  100 200 400 800 1600
Compares:  4950 19900 79800 319600 1279200
Ratios:  0.4950 0.4975 0.4988 0.4994 0.4997 O(N^2) error 0.0029

INSERTION sort with RANDOM data
Sizes:  100 200 400 800 1600
Compares:  2833 10392 39881 157673 643482
Ratios:  0.2833 0.2598 0.2493 0.2464 0.2514 O(N^2) error 0.0420


SELECTION sort with SORTED data
Sizes:  100 200 400 800 1600
Compares:  4950 19900 79800 319600 1279200
Ratios:  0.4950 0.4975 0.4988 0.4994 0.4997 O(N^2) error 0.0029

SELECTION sort with REVERSE data
Sizes:  100 200 400 800 1600
Compares:  4950 19900 79800 319600 1279200
Ratios:  0.4950 0.4975 0.4988 0.4994 0.4997 O(N^2) error 0.0029

SELECTION sort with RANDOM data
Sizes:  100 200 400 800 1600
Compares:  4950 19900 79800 319600 1279200
Ratios:  0.4950 0.4975 0.4988 0.4994 0.4997 O(N^2) error 0.0029


BUBBLE sort with SORTED data
Sizes:  100 200 400 800 1600
Compares:  99 199 399 799 1599
Ratios:  0.9900 0.9950 0.9975 0.9988 0.9994 O(N) error 0.0029

BUBBLE sort with REVERSE data
Sizes:  100 200 400 800 1600
Compares:  4950 19900 79800 319600 1279200
Ratios:  0.4950 0.4975 0.4988 0.4994 0.4997 O(N^2) error 0.0029

BUBBLE sort with RANDOM data
Sizes:  100 200 400 800 1600
Compares:  4935 19809 79239 319275 1278900
Ratios:  0.4935 0.4952 0.4952 0.4989 0.4996 O(N^2) error 0.0044


QUICK sort with SORTED data
Sizes:  100 200 400 800 1600
Compares:  488 1173 2742 6279 14152
Ratios:  0.7345 0.7673 0.7930 0.8139 0.8310 O(N log N) error 0.0376

QUICK sort with REVERSE data
Sizes:  100 200 400 800 1600
Compares:  638 1543 3748 8964 20968
Ratios:  0.9603 1.0093 1.0840 1.1619 1.2312 O(N log N) error 0.0787

QUICK sort with RANDOM data
Sizes:  100 200 400 800 1600
Compares:  672 1472 3322 7629 17655
Ratios:  1.0115 0.9629 0.9608 0.9888 1.0367 O(N log N) error 0.0258
```

SortAnalyzer counts compares for 4 different algorithms and 5 sizes for each algorithm with 3 different types of data. It then outputs the sizes, compares, ratios, BigO notation, and error.