

CIS 212 Assignment 5: 100 points

The goal of this assignment is to provide experience with reading data from a file, writing data into a file, reversing data, sorting data, and evaluating the efficiency of the sorting algorithms. Note that, for all the paths of the files in this assignment, you should use the “relative” paths, rather than the “absolute” paths, in your Java program, because your directory structure in your own computer is often different from that of others (e.g., the graders) and using the “relative” paths will avoid this issue.

1. [10] Download the following file into the same directory as your Java program:

https://classes.cs.uoregon.edu/18S/cis212/examples/phonebook_test.txt

Write your Java program to read all the data from the above file into an ArrayList (which will be referred to as the “original” ArrayList hereafter) of the phonebook entries, where each entry consists of separate fields for the name and the phone number (e.g., a class with private String variables would be considered good). The file is a plain text file, where every line has the following format: “PhoneNumber LastName, FirstName”.

2. [10] Implement a method that takes the original ArrayList of the phonebook entries and a String as inputs, and output all the phonebook entries which contain this input String in either the first name or the last name into another plain text file named “Output.txt” in the same directory as your Java program. Every line of this file should follow the same format as above. For example, if the input String is “new”, then the following should be the content of your “Output.txt”:

```
754270500 Eh, Ainew
592030468 Fannew, Ouf
880526690 Wounoot, Onew
752360287 Bussok, Neinnew
865147504 Eipysh, Wainnew
213798053 Anouh, Annew
682280707 Cheifoh, Ainew
607750213 Zeivam, Onew
317288872 Faunt, Dinnew
```

If the input String is not contained by any name, then your “Output.txt” is an empty file.

3. [20] Implement a recursive method (i.e., a method that invokes itself) which takes an ArrayList of phonebook entries as an argument and reverses the entries in this list. For example, assuming you read the list in the above picture into an ArrayList and invoke your method on this ArrayList, then after your method returns, this ArrayList should be as follows:

317288872 Faunt, Dinnew
607750213 Zeivam, Onew
682280707 Cheifoh, Aineu
213798053 Anouh, Anneu
865147504 Eipysh, Wainneu
752360287 Bussok, Neinneu
880526690 Wounoot, Onew
592030468 Fanneu, Ouf
754270500 Eh, Aineu

This is just a visualization. Here, unlike the previous two tasks of this assignment, you do not have to read data from a file and output data into a file.

4. [20] Implement a method that takes the original ArrayList of the phonebook entries as an argument and returns a sorted copy of the list using Selection Sort to sort alphabetically by phonebook last name. Your implementation must not modify the original ArrayList. You should implement your own sorting code here, and not use the Collections sort method. Cite any source used in the comments of your code.

5. [20] Implement a method that takes the original ArrayList of the phonebook entries as an argument and returns a sorted copy of the list using Merge Sort to sort alphabetically by phonebook last name. Your implementation must not modify the original ArrayList. Again, you should implement your own sorting code here, and not use the Collections sort method. Cite any source used in the comments of your code.

6. [10] Implement a method that takes an ArrayList of the phonebook entries as an argument and returns true if the input list is already sorted alphabetically by phonebook last name (false otherwise). Your implementation must not modify the input list. Use this method to test your sorting implementations.

7. [10] Implement a method that invokes your own Selection Sort and Merge Sort codes, and report the elapsed *sorting* time in seconds, excluding the time of reading data from the file. Your output should look something like (but may not necessarily be):

Selection Sort: 27.094

Merge Sort: 0.433

Please zip your Java source file(s) into a zipped file, rename that file as <Your Full Name>Assignment5.zip, e.g., SteveJobsAssignment5.zip, and then upload that file to Canvas. You are free to use whatever IDEs you prefer to complete your assignment; but no other files, except .java files, should be included in your zipped file. No “package” keyword should exist in your codes.