

Homework 5: Sorting and Searching

Introduction

For this assignment you will implement three methods (2 for sorting, 1 for searching) such that they pass the unit tests contained in the JUnit test file **SortersTester.java**. You will write all three methods in the same class file, which has been provided for you (Sorters2120.java) with stubbed implementation.

Just to get you started, I have provided you with an implementation of the bubble sort algorithm. Your job is to implement the selection sort algorithm, the merge sort algorithm (recursively), and the binary search algorithm (recursively).

For this assignment, you will be required to write Javadoc-style documentation for all of your methods, including the test methods.

Procedure

You have been provided with a class, Dog.java that implements the Comparable interface, thereby providing a compareTo() method to compare two objects. You have also been provided with a Sorters2120.java file that contains not only a fully implemented bubble sort method, but also stubbed implementations of the other methods you will be required to implement.

- 1) Implement the selection sort algorithm discussed in lecture
- 2) Implement a recursive version of the merge sort algorithm discussed in lecture
- 3) Implement a recursive version of the binary search algorithm discussed in lecture as a helper for a method to find the "indexOf" a particular element. (It should return -1 if the element is not found)

(Bonus and submission/grading guidelines on next page)

BONUS 1) (15 pts) – fix the implementation of bubble sort so that, if an entire pass of comparisons has occurred with no swaps, the algorithm quits.

BONUS 2) (20 pts) – overload the methods so that they can take a List, and a comparator object, rather than insisting that the elements of the list be subtypes of Comparable.

Submission

You will add, commit, and push your program to Gitlab. Label your homework folder "HW5" in your repository. If you choose to attempt either bonus, implement these in subfolders called "bonus1" and/or "bonus2" (each directory should contain a COMPLETE implementation of the bonus and any and all testers).

Grading

Each method will be worth 33% of your grade. You get 1 point for free. 😊