Programming on Purpose (CSC 120)
Winter 2024

Lab 8 – Oops, I did it again

2/29/2024

Objectives

- Practice recursion

Preliminary Setup

1. Create a folder for this lab.
2. Download the starter code and save it in the new folder.

You will write all of your code in `ListProcessor.java`. For inspiration and as examples of recursive methods, this file also contains the recursive implementations of linear search that we discussed in class.

To test your implementation, run `ListProcessorTester.java`.

1 Binary search

Using the pseudo-code we discussed in class, implement the recursive method for binary search.

2 Finding the index of the minimum element

Complete the method `getMinIndex` by implementing a recursive strategy for finding the minimum element in an array and returning its index.

1. Figure out the base case. Hint: What does a list look like where it is trivial to determine the minimum element?
2. Figure out the recursive case. In the recursive case you are handing a "smaller" version of the problem to a trusted friend. Once that friend comes back with an answer, what do you do with that answer?
3. Remember the pattern for recursive methods that you have seen in class: We have both a public and a private method. The private method is the one that is actually recursive and it has one or more extra parameters that change with each recursive call. Those parameters are the ones that will tell you when you've reached the base case. The public version is the coherent one that maintains information hiding. It doesn't recurse, but instead calls the private version to start the recursion going.

Write, test, and debug `getMinIndex` until it is working. Feel free to add more tests to `ListProcessorTester` if you want.

3 Sorting a list

The method `selectionSort` is intended to sort an array using a sorting algorithm called *selection sort.* Here is a recursive algorithm for selection sort. It sorts an array starting at index i:

- Check whether there is any array left to sort beyond index i.
- If so, find the index of the minimum element in the array starting at index i. (Note that you have already implemented a method for doing this step.)
- Swap that minimum value with the value at index i.
- Ask your friend to sort the remainder of the array for you starting at index $i + 1$.

Hint: This method sorts the array *in place.* I.e. it modifies the given array. That's why the return value of the method is `void`.

4 Revise your code

Before turning in any program in this class, remember this mantra:

> Just because it works doesn't mean it's good.

Part of your grade will also come from things like how understandable and readable your code is. You will also be graded on the neatness, presentation, and style of your program code.

Make sure that all modules and functions are documented (even those that you didn't write).

Don't forget to cite who you received help from and include the honor code affirmation in the javadocs of each class that you wrote or modified:

```
I affirm that I have carried out my academic endeavors
with full academic honesty. [Your Name]
```

5 How to submit

Submit the project by uploading `ListProcessor.java` to Gradescope.

Make sure to add your partner to the submission so that you both have access to it through Gradescope.