Recursion exercises

Important note

In addition to this very short set of programming exercises, there is a set of questions to answer in a separate Gradescope assignment. That is, there are a total of two "half" Gradescope assignments this week, a "Part A" and a "Part B". They are equally weighted, and will count together for a single assignment when computing your class grade. Part A includes questions on both running time (aka time complexity) analysis and recursion. The programming assignment (Part B) includes four short methods to write, three of which must be implemented using recursion.

Overview

In class week, we reviewed the concept of recursion that you've seen before in 160. In this assignment, you will implement three recursive (and one iterative) method on simple inputs. The recursive methods are – we hope! – methods you could implement iteratively; the point of this assignment is to give some practice with recursion.

We've provided a set of unit tests to help with automated testing, though you might also want to write a class with a `main` method for interactive testing. As before, we've disable the timeout code so you can use the debugger, but if your code gets stuck during testing, you might want to uncomment these two lines at the top of each test file:

```
@Rule
public Timeout globalTimeout = Timeout.seconds(10); // 10 seconds
```

Goals

- Translate Javadoc descriptions of behavior into code.
- Practice implementing recursive methods.
- Test code using unit tests.

Downloading and importing the starter code

As in previous assignments, download and decompress the provided archive file containing the starter code. Then import it into Code in the same way; you should end up with a `recursion-exercises-student` project.

What to do

As usual, look over the files we've provided. The `support/` directory (which contains files you should *not* modify) contains the `Node` class you've seen in lecture. the `src/` directory contains a single file with four methods for you to implement. Finally, the `test/` directory contains tests of the desired behavior.

Manually graded items

There are four manually graded items. In particular, the human graders will check that each of your methods is correct *and* implemented using the specified paradigm (iteratively or recursively). This is mostly a set of recursion exercises – only `sumOfDigitsIterative` should be implemented iteratively, the others should use only recursion! Both of these conditions must hold for you to receive the credit each manually graded item; there is no partial credit on these items.

Submitting the assignment

When you have completed the changes to your code, you should export an archive file containing the entire Java project. To do this, follow the same steps as from Assignment 01 to produce a `.zip` file, and upload it to Gradescope. Note that if you want things to upload faster, you can use an external program to zip only the `src/` directory by expanding the project; that's all this autograder requires.

Remember, you can resubmit the assignment as many times as you want, until the deadline. If it turns out you missed something and your code doesn't pass 100% of the tests, you can keep working until it does. We will not do any manual grading until after the late deadline for the assignment passes.