Ben Bartels

WIP Report

I decided to take a TDD approach to this project because I think it makes a lot of sense for this particular project. I know the methods that I need to implement, what they’re called and how they should behave, so it’s pretty simple to come up with the test cases prior to writing the code for each method. So far I have written a few tests for two constructors (empty and taking in an int), my toString method, and my addition method. I have also implemented those methods in the class itself.

Beyond those methods, nothing works yet. It took me a while to decide on how I wanted my class to work internally. I wasn’t sure what type to make my array or how I wanted to handle negative numbers. I finally decided to go with an array of 32-bit integers. I don’t allow an integer in the array to go above 999999999 because this makes the parsing to and from strings really simple, as well as addition. I know it’s not the most efficient use of space, but I wasn’t sure how else to make it work. In the future, I might change this if I come up with a better idea, or if you have advice to give me in your feedback.

I have a lot of different operations I’d like to implement in this library, such as modulo division and factorials, but my focus is on completing the four main operators. A full list of my ideas so far can be found in the source code comments at the top of my class. I also still need to create some kind of driver program. I don’t have any ideas yet for one beyond a console program that allows the user to do simple calculations with two big numbers. Moving forward, I will be working on subtraction, multiplication, and then division. I think I know how subtraction and multiplication will work (worst case scenario, my multiplication method can call the addition method a huge number of times), but I haven’t figured out division yet. I haven’t thought about any of those enough yet though to know for sure. I also need to create a lot more tests for the methods that I have implemented and try to refactor my current implementations because I have a lot of errors right now (some of them are noted in comments in the source code).

One thing I’m really struggling with right now is how to determine the size of the resulting array for an operation before doing the operation itself. I don’t want to have a bunch of wasted space in my array, but I also don’t want to have to resize it after the fact. This problem in particular is something I hope to fix prior to my final submission. If I can’t figure it out on my own, I’ll probably have to look up open-source implementations and see how other people solved that problem.