Austin Choi

CS 1632 – DELIVERABLE 4

PROPERTY-BASED TESTING

https://github.com/M94/cs1632-d4

For project 4, I chose to perform property-based testing on Java’s Arrays.sort() method because I wanted to practice writing more test code and gain more experience in working with testing libraries. For this particular deliverable, I wrote a Java-based test class using jUnit and the Java QuickCheck property-based testing library.

My test class checks make sure that the Arrays.sort() method upholds four different invariants. Firstly, when the function takes an input array, it should output an array of the exact same size. Secondly, the sort function must be pure, meaning that it should produce the same output array when run on two copies of the same array. With that in mind, the function should also be idempotent and produce the same sorted array when run once or twice on a single array. And finally, each successive element in the sorted array should have a value greater than or equal to the previous element’s value.

For each test, my program generates ten unique arrays of different sizes for three data types: characters, integers, and strings. There are four different property tests, so a total of 120 unique arrays are tested. I used the QuickCheck library to generate a wide range of data for my arrays.

One issue I ran into was implementation. I wanted to write concise code that tested different array types, but since Java is strictly typed, I could not just lump every array into one list and write a single for loop to test every array. I had to store the arrays of each data type into separate multidimensional arrays and write unique statements for each one. Despite this shortcoming, I did my best to avoid repetitive code.

After finishing Deliverable 3, I have become more aware of the implementation and usefulness of property-based testing. I noticed that I did not have to write that much code to check for important and desired traits. And to test all these different traits, I would have only needed to use a single set of arrays for my input (though my program generated new array values for each test). For the amount of code written, I think that property-based testing will definitely be a useful tool for me in the future as an engineer.