Members: Gregory Hartranft (gdh12@pitt.edu)

CS 1632 - DELIVERABLE 4

PROPERTY-BASED TESTING

Github: <https://github.com/Sindalf/CS1632-HW4>

[Date]

[Company name]

[Company address]

This project I choose Property based testing using Arrays.sort(int[]). I tested three different properties, Purity, all values contained, and increasing property. I also added my own sort function as a way of testing since a sort function is an algorithm that has one specific output. This way we know that regardless of the algorithm the output should be exactly the same. There were very little struggles in creating these tests. The arrays were created using (Math.random() \* 100) for the size and values of the array. This also makes 0 inclusive so 0 should also be tested but is rather unlikely. This also makes input value properties held true by default due to java's strong typing otherwise it wouldn’t compile or we would crash.

When testing purity, it was also similar to testing if its idempotent or not. This is because when testing I compared it against a bubble sorted array. If the two arrays stay equal after one sort and after a second sort with no alterations to the bubblesorted array then we know that both purity and idempotent holds true. I also tested if every value in the original array was also in the sorted array using the binarysearch() method. Every single value is checked for the length of the sorted array which in turn would also hold the property of "no element not in the array". Lastly I checked if the values were always increasing or stayed the same as the array went on. Since you can put the check in the assertion we know then that if this passes then there is zero possibility for decreasing values so this property would also hold true.

This all leads me to a question, is it proper to say that other properties are held true since due to the way you test one or due to the format of the code that it couldn’t possibly be anything else. I guess that isn't proper testing but as we went over exhaustive testing is not exactly possible so I'm guessing that this would be ok in the "real world" but not ok if you are looking to formally test something.

