**CS 1632 - DELIVERABLE 4 - PROPERTY-BASED TESTING**

Benjamin Nimchinsky

src/LaboonifyClass.java

test/LaboonifyClassTest.java

https://github.com/znaeb/CS1632QA-Diliverable4

Summery:

I was asked to say why I chose this project, and while that might be a good question for a different assignment, I do not believe it is a good question for this assignment. I didn’t choose this project, it was assigned to the entire class, and unless I have completely misunderstood the requirements to this assignment (also a valid possibility), the only real choice I had regarding this assignment’s requirements was which properties to test the Laboonify method with. As such I will, possibly erroneously, assume the question is asking about my choice of properties to test.

The first property I tested was that the output array’s size was one more than the input array’s size. First I generated my 100 initial arrays of varying sizes and contents, and stored them in an array (an array that stores 100 integer arrays), (although I could have just as easily tested each array as it was generated, storing the arrays would potentially be beneficial for data checking). I then looped through the outer array to get each individual array’s size, and compared it to the size of the array generated by the Laboonify method (given the current array as an input). If the Laboonify Method’s output didn’t have one more element than its input, the test failed.

The second property I tested was that the output array properly squared each element. First the 100 arrays were generated and stored. Then each array was compared index by index with its Laboonified counterpart, excluding the Laboonify’s Last index, to see whether the Laboonified index was the square of the original index.

The third property I tested was whether the last bit of the Laboonify method was the sum of all of the previous values. I did this in two separate tests, one test actually generated the squares of the originally inputted array, and summed them together to compare it to the last bit. However, I felt that this test was somewhat iffy in that it could also be seen as also testing the second property, thus I made a fourth test (or second test depending on how you look at it) to test the third property as well. My fourth test looked at the Laboonified array, and directly summed up all of the indices (excluding the last one), and checked if it equaled to the last index.

I had no real issues with this assignment, other than the fact that I am still a bit unsure if I was supposed to make my own auto-generated test cases (which I did do), or use something such as QuickCheck.

Test Results Image:

