|  |
| --- |
| Jeffrey Josephs |
|  |
| CS 1632 - DELIVERABLE 4: PROPERTY-BASED TESTING |
|  |
| 10 November 2015 |

Summary

The option that I chose for Deliverable 4 was property-based testing because I was intrigued by how unit testing worked with Junit and I wanted to further learn how it could be utilized. In terms of property-based testing, I was not as familiar with the concept in a formal sense, only really knowing informally how the function should work, but now was able to observe formal properties of the sorting function. When creating my tests, I chose to implement my unit tests for the properties pertaining to the output array being the same size as the input array, every element in the input array being in the output array, idempotent, running the sorted array again should not result in any changes, and pure, running the sorting function on the input array twice and the need for both outputs to be the same.

When going about my testing, I first had to create a random number of arrays with a random number of random integers so that I could run the sorting function on various data samples. From there, I could iterate through all of the arrays while running the sorting function on them and keeping track of the original, pre-sorted array so that I could do the comparison or run it through the sorting function a second time. After the comparison, I mainly used the assertEquals, and because I had an assertion for each array sort, one failure would catch even if it were one out of 100 samples.

I do not feel that I experienced any problems while creating the tests, but I definitely feel that I have a better understanding of property-based testing. I now understand in a formal sense to test properties of a function, and it has allowed me to apply those to other functions where I can further test my code.

GitHub repository: https://github.com/jhj17/CS1632-Deliverable4

Executed Unit Tests

