Sean Penney
Bug/progress report

First off, the bintree_binary_tree.tstl and bintree_rb_tree.tstl tests are functioning correctly. The performance tests for the binary tree and red-black trees current have a bug, however. In the performance tests there seems to be an issue with assigning 0 to the insertcalls variable and increment by 1.

One bug found in the system under test occurs when the method to do an inorder traversal on the binary tree is called. Calling this method results in an AttributeError exception in certain cases.

Another bug found occurred when a single element was in the binary tree, and the method to find the minimum value was called. This resulted in a ValueError('Tree is empty') exception, even though there was an element in the binary tree. I was even able to replicate this bug in the python interpreter, using the reduced test steps that tstl printed out.

One bug found appears to be a false alarm, as a guard in the tstl code is not being checked properly. A guard is in place to make sure that a key is in the data structure before trying to remove the key. However, the remove method was called on a key that is nonexistent.

Another bug turned out to actually be a problem with the tstl file. The max method is called on both the bintrees BinaryTree and the dict I am using for reference. The assertion failed, however, this is because different values were inserted in the steps. I was confused as to why the bintrees BinaryTree and the reference did not previously insert the same values.

My test system has improved, however there are some issues to address. Some of the methods I wanted to test included the set methods. I will need to create two objects for the bintrees data structures, and two objects for the dict. The set methods can then be called for the bintrees objects, and for the dict objects. The results of the set methods can then be compared. I should be able to accomplish this before the end of the term. Other progress that should be made before the end of the term is fixing the problems with the tstl files that test for performance.

Given that many of the bugs were actually problems with the tstl file it is hard to evaluate the quality of the system under test at this point. At least one bug does appear to be a problem with the system under test itself. Code coverage was very minimal however (approximately 1%). Increasing code coverage by the end of the term would ideally lead more bugs found in the system under test rather than bugs in the tester. Based on the pypi.python.org site, there were 4412 downloads in the last month for bintrees, which makes it plausible that the library would have some bugs.