CS562: Applied Software Engineering

Part 3: Bug/progress report

Omkar Thakur
Email: thakuro@oregonstate.edu

Describe some bugs you found:

By testing PyBST library using TSTL, no bugs were found so far. But I think compatibility issues were found between some functions.

Explain progress to date:

The main function BSTree was tested thoroughly tested. The main functions for other trees such as AVLTree(), SplayTree(), RBTree() were also thoroughly tested. The keys and values were inserted in it and it kept on running. As a result, it generated 2477790 test operations in 37 seconds . Out of which 24778 operations were executed. The execution had to be stopped by an keyboard interrupt. As the values were generated, they were also deleted. But still, no bug was found. The quality of sut is very good in such default or basic cases.

The feature of insert_from was tested on the BSTree() function. Due to which the running time was just 0.001 sec.. A total of 8 test operations were done, out of which 1 was executed. This function was not executed in a proper manner. Therefore, alternative implementations will be discussed for this.

The feature of deletefrom was also tested n this. This feature was not tested appropriately because, of the node.left was not implemented appropriately. The total runtime was 0.0009. 1 was executed with 6 total test operations.

The feature of getheight was tested. The test case was successful. No bugs were found in this feature. Additional testing will be done of this feature with other features such as switchnodes(). The above explanation is for all four trees.

Minor functions such as get_min(), get_max(), inorder(), preorder(), postorder(), levelorder() were tried for testing. But there were some problems in syntax.

The function switchnodes() was not tested properly, since only 8 test operations were executed.

Estimate your progress at the end of term:

The functions of

1. insert_from()
2. BSTree(seq)
3. AVLTree(seq)
4. SplayTree(seq)
5. RBTree(seq)

6. delete_from()
7. switch_nodes()
8. getnode()

will be tested thoroughly.

Some minor functions such as

1. get_min()
2. get_max()
3. inorder()
4. preorder(),
5. postorder(),
6. levelorder() will also be tried for testing. But more emphasis would be given on the major functions. Performance testing will also be done between the four trees.

Discuss quality of the SUT

PyBST is a well tested library with limited documentation. However I believe that the developers of this library have not taken novice users into account. This is because the documentation is not complete. Due to which the novice users may make mistakes considering the compatibility of the functions. I believe the Tree(seq) function is not compatible with some of the functions. This is not that big library if you only consider only bstree.py . But since, I am also testing avltree.py, rbtree.py and splaytree.py . The amount of code is more. Most of the functions are very well tested. I believe this library is very well tested. But I think due to non compatibility of small functions, there exists novice user misuse.

Code Coverage

The test code coverage of the PyBST code is almost 15%. Most of the main functions have been tested in some circumstances. They are yet to be tested in various circumstances. I think the coverage was less because I am testing more number of python files (rbtree.py, splaytree.py, avltree.py and bstree.py) In total it was equivalent to 1300 lines of code.

References:

https://pypi.python.org/pypi/**pybst**/1.0

https://github.com/TylerSandman/py-bst/tree/master/**pybst**

https://github.com/TylerSandman/py-bst/blob/master/**pybst**/draw.py

https://anaconda.org/auto/**pybst**

https://github.com/a**groce/tstl**