CS 562 Project Proposal

Omkar Thakur

pybst library

About:

The following library implements Binary Trees, AVL Trees, Splay Trees and Red Black trees in python. The PyBST also provides a module for plotting these trees using networkx and matplotlib.

Creator:

Tyler Sanderson

System can be found at:

 https://pypi.python.org/pypi/pybst/1.0#downloads

Development Status of Library:

Stable

Programming Language Compatible:

Python 2.7

Operating System going to be used for testing:

Ubuntu 14.4.03

Contents in Library:

The tree classes provided are as follows:

BSTree – represents an unbalanced Binary Search Tree

AVLTree – represents a balanced AVL Tree

SplayTree – represents an adjusted Splay Tree

RBTree – represents a balanced Red Black Tree

The constructors used are as follows:

Tree() - > Creates a new empty tree

Tree(seq) -> Creates a new empty tree from seq [(key1,val1),(key2,val2),…,(keyn,valn)]

What I wish to test:
The testing involved in this product will mostly consist of automation testing. This is because of using TSTL. This library is not compatible with python 3. Performance testing will also be done with various trees involved in this library. The compatibility of TSTL may remain an issue since it is still in its beta. But maximum test cases will be made. The plotting function of this library will also be tested. The results of the tests will be compared with the different trees. A Comparison will be made in the final report. The bugs will be reported. There will be with various options available in random tester. Testing under various options such as flags will also test this software in various conditions. The thinking required by tester will be out of box. This is because inorder to find as many bugs as possible .This can only be done innovatively. As, finding regular bugs will not be of use. Bug reports will be written which will assist the developers of TSTL. This library also provides data structures, which is designed to work with "relational" data. Each of the following methods will be tested with all of the trees and they will be compared. This library also plots the trees. Therefore two dependencies are required for plotting. They are networkx and matplotlib. The plotting of the various trees will also be tested in various circumstances. This library is under the license of GNU GPL 3

The PyBST library has a number of tree methods available. Such methods are useful for insertion, deletion,changing the order of the tree, deleting the key. Therefore testing these features is vitqal in case of different trees available.

The following are the valid tree methods which I wish to test:

is_valid() -> Produces True if Tree is a valid tree of its type, else False.

preorder() -> Produces a sequence of the Nodes in Tree in preorder.

inorder() -> Produces a sequence of the Nodes in Tree in inorder.

postorder() -> Produces a sequence of the Nodes in Tree in postorder.

levelorder() -> Produces a sequence of the Nodes in Tree in levelorder.

get_node(key) -> Produces the Node in Tree with key attribute key.

insert(key,val)   <==> Tree[key] = value. Inserts a new Node with key attribute key and value attribute val into Tree

insert_from(seq) ->   Inserts keys and values from seq [(key1,val1),(key2,val2),…,(keyn,valn)] into Tree.

get_max()   <==> max(Tree). Produces the Node with the maximum key in Tree.

get_min()   <==> min(Tree). Produces the Node with the minimum key in Tree.

get_element_count()   <==> len(Tree). Produces the number of elements in Tree.

get_height()   -> Produces the height of Tree

delete(key)   <==> del Tree[key]. Deletes the Node with key attribute key from Tree.

delete_from(seq)   -> Deletes Nodes with keys from seq [key1,key2,…,keyn] from Tree.