

Simulated coursework exercise - Binary Search Tree

An often used data structure in computer science is the binary tree structure. This is a graph-like structure where the values below each branch in the tree are sorted according to a comparison with the value at a particular node.

For example, one might have a tree where each value on the left of a node is less than the current value, and each value on the right is greater.

More information about binary search trees:

https://en.wikipedia.org/wiki/Binary_search_tree

Your task is to **implement a binary search tree** using the python language. You must use version **python3.4** or higher for this task.

Your solution should implement the following interfaces:

A file named "tree.py":

- Contains a class "Tree" with methods:
 - `__init__(self, value)` - constructs a new tree with this value as the root
 - `insert(self, value)` - inserts a value to the correct location in the tree. It returns true if the item was inserted, and false if it already existed in the tree.
 - `find(self, value)` - this looks through the tree and returns the depth of the value if the value is contained in the tree, or -1 otherwise. The root is depth 0.

As long as the described interface is met, you may use any internal representation or extra methods as you like.

Some test files. These should follow the standard unit testing framework offered by python 3. You should use these test files to check your own solutions, but remember that they will be used on other people's solutions:

- They should assume that the file "tree.py" is in the same directory.
- They are only required to test what is offered by the interface above.

Within the coursework, you will also find some skeleton files that offer the interfaces, for you to fill in yourself.