Week 6 Assignment

Submit Assignment

Due Monday by 6:40pm **Points** 10 **Submitting** a text entry box or a file upload

Please make sure that you follow the submission guidelines to avoid having points deducted.

1. **Using the attached, write the following methods. (Edit linked_binary_tree.py)**
   - ***delete_subtree(p)***, that removes the entire subtree rooted at position p, maintaining the integrity of the data structure. Note inside your program as a comment the running time of your implementation
   - ***swap(p,q)***, that has the effect of modifying the tree so that the node referenced by p takes the place of the node referenced by q, and vice versa. Do not swap the elements. Please note that p's children become q's children and vice versa and they swap parents. **Week 6 Assignment Picture-1.pptx**
   - ***sizeOf(p)*** that returns the number of nodes in the tree rooted at p
   - ***elementsAtLevel(k)*** that returns the elements at level k (Hint: Look at __repr__ in linked_binary_tree.py)
   - ***levelOf(e)*** that returns the level of element e or -1 if the element is not in the tree.

Binary Tree Code.zip

2. **Implement a binary search tree using the array-based representation described in Section 3.2. Implement the following methods. Indicate the worst case running time of each of the methods. (Probably best to write a completely new class and not reuse any code from the first question)**

Integer __len__()	The number of values added to tree
add(e)	Adds e to the tree maintaining the binary search tree property
Boolean contains(e)	Returns True if e is in the tree, False otherwise
__repr__()	Returns a string representation of the tree showing the tree structure LinkedBinaryTree
Iterator inorder()	Returns an Iterator for the indorder traversal of the tree

Integer height()	Returns the height of the tree