Algorithms and Data Structures

Summer Semester 2024

For discussion on Wednesday, June 5, 2022

1. Write a self-contained implementation of a linked binary tree class in Python. This should be done in the spirit of Exercise 5 Problem 1, i.e., without using abstract base classes or an abstract position class. You should implement the following methods: `T.root()`, `T.parent(v)`, `T.left(v)` (return left child), `T.right(v)` (return right child), `len`, `T.is_empty()`, `T.is_leaf(v)`, `T.add_root(e)`, `T.add_left(v,e)` (error if there is already a left child at node v, otherwise, if e is a tree, attach it as the left subtree to v and reset it to an empty tree, if e is any other Python object, create a new leaf node as the left child of v with element e), `T.add_right(v,e)` (likewise for right child), `T.replace(v,e)`, `T.delete(v)` (error if v has two children, otherwise remove v from the tree, attaching its child, if there is one, to its parent and returning the element stored in v).

2. (GTG Exercise R-8.5) Describe an algorithm, relying only on binary tree operations, that counts the number of leaves in a binary tree that are the left child of their respective parent.

3. (GTG Exercise R-8.20) Draw a binary tree T that simultaneously satisfies the following:

   - Each internal node of T stores a single character.
   - A preorder traversal of T yields EXAMFUN.
   - An inorder traversal of T yields MAFXUEN.

4. (GTG Exercise of R-8.23) Let T be an ordered tree with more than one node. Is it possible that the preorder traversal of T visits the nodes in the same order as the postorder traversal of T? If so, give an example; otherwise, explain why this cannot occur. Likewise, is it possible that the preorder traversal of T visits the nodes in the reverse order of the postorder traversal of T? If so, give an example; otherwise, explain why this cannot occur.