**Mason Shepherd**

**Homework 5: due February 27th 11:59PM.**

R-8.5 Design an algorithm (suggest to use the Binary Tree operations) to count the number of leaves in a binary tree that are the *left* child of their respective parent.

Ans:

**Algorithm** countLeftChild(*p* {where *p* is the root of the given binary/subtree}):

*count* = 0

**for** each child *c* in children(*p*) **do**

*count* += countLeftChild(*c*)

**if** isExternal(*p*) **AND** *p* == left(parent(*p*)) **return** 1

**else**, **return** *count*

R-8.12 Draw the binary tree representation of the following arithmetic expression: “(((5+2)∗(2−1))/((2+9)+((7−2)−1))∗8)”.

Ans:

R-8.22 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.

Ans:

C-8.28 The ***path length*** of a tree *T* is the sum of the depths of all positions in *T*. Describe a linear-time method for computing the path length of a tree *T*.

Ans: (??????)

**Algorithm** pathLength(*T*, *p*, *d*):

**if** *p*.isExternal() **return** *d*

**else**

depth = *d*

**for** each child *c* of *p* **do**

depth = depth + pathLength(*T*, *c*, depth+ 1)

**return** *d*