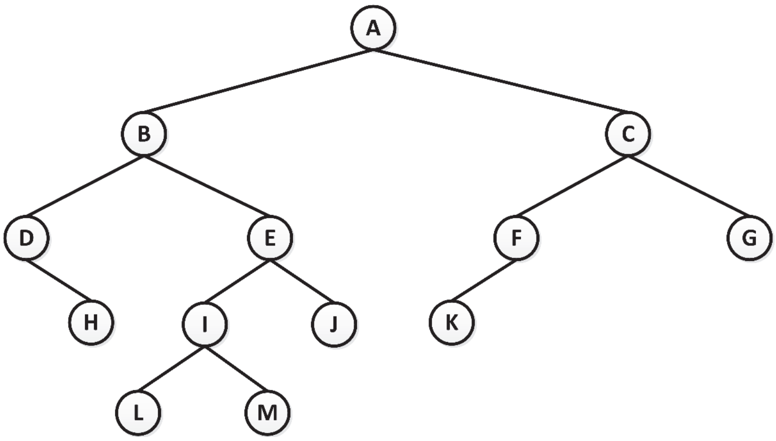
Small Group Tutorial 2, 27/2 - 3/3/2017

1. Let T be a *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, argue why it cannot occur.

   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, argue why it cannot occur.

2. Let T be the binary tree as below



   (a) Give the output of toStringPostorder(T,T.root()) method presented below.

```
public static String toStringPostorder(Tree T, Position v){
        String s = "";
        for (Position w: T.children(v))
           s += toStringPostorder(T, w) + ", ";
        s += v.element().toString();        // main visit action
        return s;
}
```

   (b) Give the output of toStringPreorder(T,T.root()) method presented below.

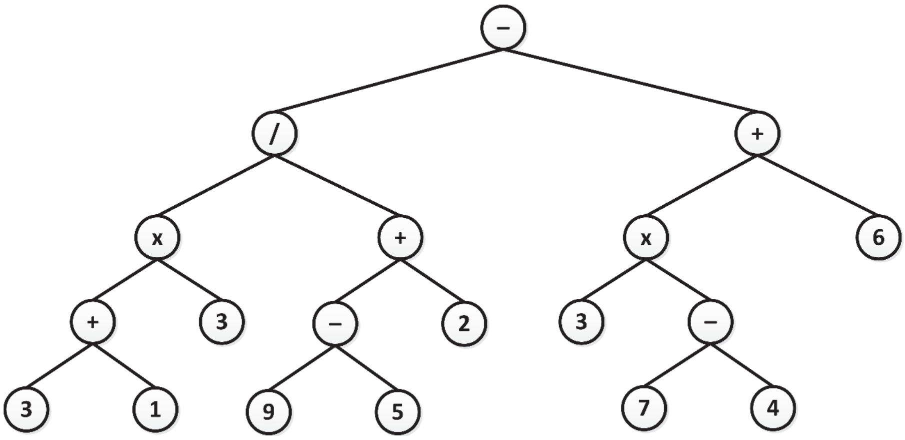
```
public static String toStringPreorder(Tree T, Position v){
        String s =  s += v.element().toString();        // main visit action
           ;
        for (Position w: T.children(v))
           s += ", " + toStringPreorder(T, w);

        return s;
}
```

3. Draw a (single) binary tree T such that:

   - Each internal node of T stores a single character
   - A preorder traversal of T yields E X A M F U N
   - An inorder traversal of T yields M A F X U E N

4. Let T be the binary tree as below.
   Give the output of printExpression(T,T.root()) algorithm presented below.

```
Algorithm: printExpression(Tree T, Position v)
    if T.isInternal(v) then
        print "("
    if T.hasLeft(v) then
        printExpression(T, T.left(v))
    if T.isInternal(v) then
        print the operator stored at v
    else
        print the value stored at v
    if T.hasRight(v) then
        printExpression(T, T.right(v))
    if T.isInternal(v) then
        print ")"
```



5. Proof the following properties of binary trees (for $n \geq 1$)
   (a) $1 \leq n_e \leq 2^h$
   (b) $h \leq n_i \leq 2^h - 1$
   (c) $\log_2(n+1) - 1 \leq h \leq n - 1$

   where n – number of nodes; n_e – number of external nodes; n_i – number of internal nodes; h – height of a binary tree

6. ADDITIONAL – Describe in pseudo–code, a nonrecursive method for performing an inorder traversal of a binary tree T.