Exercise 1: (INFO1105 and INFO1905)

1. Show a tree with \( n \) elements achieving the worst-case running time for the following algorithm:

Algorithm 1 Calculates the depth of a tree

```
1: procedure DEPTH(Tree T, node v)
2:   if v is the root of T then
3:     return 0
4:   else
5:     return 1 + DEPTH(T, w), where w is the parent of v in T
6:   end if
7: end procedure
```

Also, show a tree that achieves the best-case running time.

Answer

(a) Consider a degenerate case, i.e., a binary tree where for every internal node there is only one child.

(b) Consider a complete binary tree.

2. Let \( T \) be a 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 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, argue why this cannot occur.

Answer

It is not possible for the postorder and preorder traversal of a tree with more than one node to visit the nodes in the same order. A preorder traversal will always visit the root node first, while a postorder traversal node will always visit an external node first. It is possible for a preorder and a postorder traversal to visit the nodes in the reverse order. Consider the case of a tree with only two nodes.
3. Draw a (single) binary tree $T$ such that
   - Each internal node of $T$ stores a single character
   - A preorder traversal of $T$ yields EXAMFUN
   - An inorder traversal of $T$ yields MAFXUEN.

Answer

![Tree Diagram]

Figure 1: Solution

**Exercise 2: (INFO1105)**

We can define a binary tree representation $T'$ for an ordered general tree $T$ as follows (see Figure 2):

- For each node $u$ of $T$, there is an internal node $u'$ of $T'$ associated with $u$.
- If $u$ is an external node of $T$ and does not have a sibling immediately following it, then the children of $u'$ in $T'$ are external nodes.
- If $u$ is an internal node of $T$ and $v$ is the first child of $u$ in $T$, then $v'$ is the left child of $u'$ in $T$.
- If node $v$ has a sibling $w$ immediately following it, then $w'$ is the right child of $v'$ in $T'$.

Given such a representation $T'$ of a general ordered tree $T$, answer each of the following questions:

1. Is a preorder traversal of $T'$ equivalent to a preorder traversal of $T$?
2. Is a postorder traversal of $T'$ equivalent to a postorder traversal of $T$?
3. Is an inorder traversal of $T'$ equivalent to one of the standard traversals of $T$? If so, which one?

![Tree Diagrams](image)

(a) $T$

(b) $T'$

Figure 2: Representation of a tree with a binary tree: (a) tree $T$; (b) binary tree $T'$ for $T$. The dashed edges connect nodes of $T'$ that are siblings in $T$.

**Answer**

1. yes
2. no
3. yes, postorder.

**Exercise 3: (INFO1105 and INFO1905)**

Give an $O(n)$-time algorithm for computing the depth of all the nodes of a tree $T$, where $n$ is the number of nodes of $T$.

**Answer**

This can be done using a preorder traversal. When doing a "visit" in the traversal, simply store the depth of the node’s parent incremented by 1. Now, every node will contain its depth.
Exercise 4: Binary tree coordinates (INFO1105 and INFO1905)

Consider a binary tree $T$ where each node has a pair of $(x, y)$ coordinates in the plane. The $x$-axis represents the order produced by inorder traversal. The $y$-axis is the height, i.e., the $y$ coordinate is the height of a node in $T$.

Write a method in `BinaryTree.java` that computes the node coordinates and outputs a binary tree to a file in the `dot` format. For example, the following dot file (`sample.dot`) defines three nodes $a$, $b$, and $c$ positioned at $(2, 0)$, $(0, -2)$ and $(0, 2)$ respectively (the coordinates are in inches by default). There are two edges $(a, b)$ and $(a, c)$:

```plaintext
graph {
    a [pos="2,0!"];
    b [pos="0,-2!"];
    c [pos="0,2!"];
    a -- b;
    a -- c;
}
```

Once you have a dot file, you can visualise your binary tree using the following command to produce a PNG file:

```
dot -Kfdp -Tpng -o sample.png sample.dot
```

![Figure 3: sample.png](sample.png)

Exercise 5: (INFO1105 and INFO1905)

Describe, in pseudocode, a nonrecursive method for performing an inorder traversal of a binary tree in linear time. (Hint: consider using a stack)

**Answer**

```
Algorithm inorder(Tree T):
```

```javascript
Data Structures Page 4 of 5
```
Stack $S \leftarrow$ new Stack()
Node $v \leftarrow T.\text{root}()$
push $v$
while $S$ is not empty do
    while $v$ is internal do
        $v \leftarrow v.\text{left}$
push $v$
while $S$ is not empty do
    pop $v$
    visit $v$
    if $v$ is internal then
        $v \leftarrow v.\text{right}$
push $v$
while $v$ is internal do
    $v \leftarrow v.\text{left}$
push $v$

**Exercise 6: (INFO1905)**

Describe a nonrecursive method for performing an inorder traversal of a binary tree in linear time that does *not* use a stack.

**Exercise 7: (INFO1905)**

Let $T$ be a binary tree with $n$ nodes ($T$ may be realised with an array list or a linked structure). Give a linear-time algorithm that uses the methods of the BinaryTree interface to traverse the nodes of $T$ by increasing values of the level numbering function $p(v)$ defined as follows:

$$p(v) = \begin{cases} 
1, & \text{if } v \text{ is the root of } T \\
2p(u), & \text{if } v \text{ is the left child of node } u \\
2p(u) + 1, & \text{if } v \text{ is the right child of node } u 
\end{cases} \quad (1)$$

This traversal is known as the *level order traversal*.

**Answer**

Algorithm levelOrderTraversal(BinaryTree $T$):
Queue $Q = \text{new Queue}()$
$Q.\text{enqueue}(T.\text{root}())$
while $Q$ is not empty do
    Node $v \leftarrow Q.\text{dequeue}()$
    if $T.\text{isInternal}(v)$ then
        $Q.\text{enqueue}(v.\text{left}_\text{child})$
        $Q.\text{enqueue}(v.\text{right}_\text{child})$