ECE 250 *Algorithms and Data Structure*
Department of Electrical and Computer Engineering
University of Waterloo
Please send any comments or criticisms to dwharder@alumni.uwaterloo.ca
with the subject ECE 250 Notes 4.01.
Assistances and comments will be acknowledged.

4.1 The Tree Data Structure

We have already seen a number of data structures:

singly and doubly linked lists, one- and two-ended arrays, and circular arrays.

We will now look at a new data structure: the rooted tree.

4.1.1 Description

A rooted tree is like a linked list, it has a *first* node, but this node is referred to as the *root* of the tree. Every node within the tree has a variable number of next pointers, and these next pointers reference other nodes within the tree. Each node within the tree, with the exception of the root, has exactly one other node that points to it.

An example of a tree is shown in Figure 1.

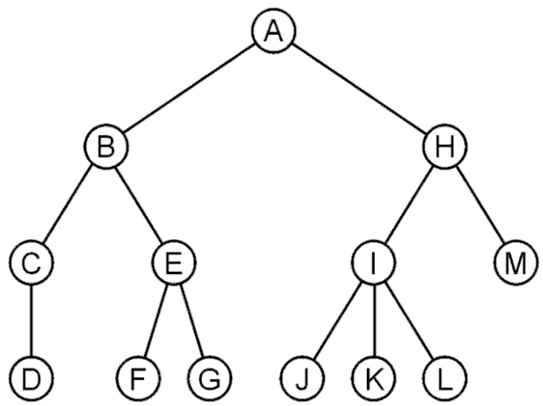

Figure 1. A tree with a root storing the value 'A'.

Convention: rather than referring to a node as "a node storing the value *X*", we will refer to such a node as "node *X*".

Convention: the root node of a tree will be referred to as the *root*.

Definitions: child, children, parent, siblings, degree, leaf nodes, internal nodes

For any node *X* within a tree, the collection of all nodes to which it points as *children of X* and each is said to be a *child of X*.
Given any node *X* other than the root, the unique node pointing to it is said to be the *parent of X*.
Two nodes that have the same parent are said to be *siblings*.
The *degree* of a node is the number of children.
Nodes of degree zero are called *leaf nodes*. All other nodes are internal nodes.

For example, in Figure 1, the children of node E are F and G, its degree is 2 (written deg(E) = 2), and its parent is node B.

The nodes D, F, G, J, K, L, and M are leaf nodes, while nodes A, B, H, C, E, and I are internal nodes.

Figures 2 shows the phylogenetic tree of the clade of carnivoramorpha. This is a special form of tree where all nodes have two children. Highlighted are the leaf and internal nodes.

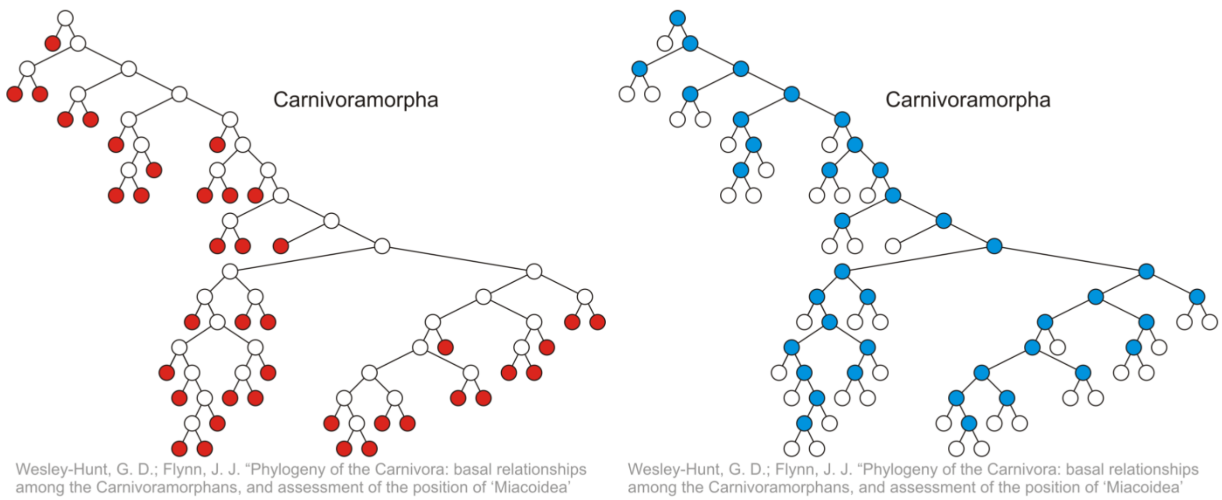

Figure 2. The phylogenetic tree of the clad of carivoramorpha.

Definitions: Ordered and Unordered Trees

When a species splits into two, there is no significance to the order of the children. Similarly, in organizations, if there are three research groups within the research and development department, there is order that can naturally be imposed order on the three groups.

For other trees, however, the order of the children may be relevant. If the children are linearly ordered, the tree is said to be an *ordered tree*. For example, the two trees in Figure 3 are identical if the trees represent unordered trees, but they are different if the trees are ordered.

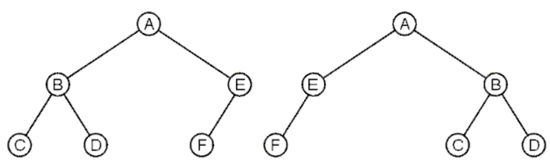

Figure 3. Two equal unordered trees but different ordered trees.

As can be seen in Figure 2, there is a single path that takes you from the root node to any node within the tree.

Definitions: Paths, depth of a node, height of a tree

The shape of a tree gives a natural flow from the root down to any node within the tree, as is shown in Figure 4.

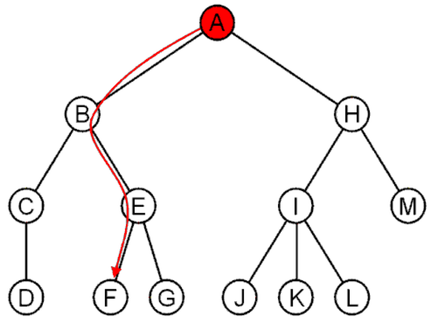

Figure 4. The natural flow from the root to any node within the tree.

A path within a tree is a sequence of nodes, (a_0, a_1, a_2, ..., a_n) where each node in this sequence is a child of the previous node in the sequence, *i.e.*, a_{k+1} is a child of a_k. The length of this path is n which can be interpreted as the number of *edges* connecting the nodes.

In Figure 5, the path (B, E, G) has length 2.

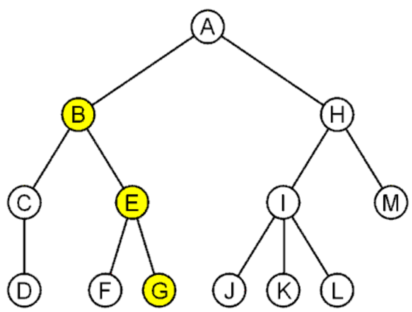

Figure 5. A path (B, E, G) of length 2.

Figure 6 has two paths, one in yellow of length 10 (containing 11 nodes) and one of length 4 (containing 5 nodes).

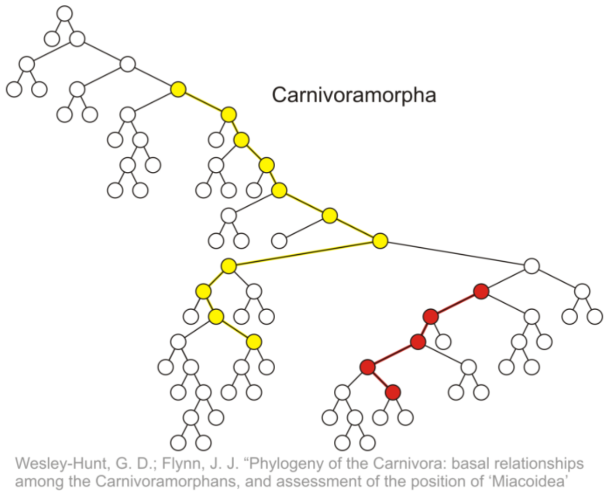

Figure 6. Two paths of length 10 (yellow) and 4 (red).

For each node within a tree, there is a unique path from the root node to that node. The length of this path is referred to as the *depth* of the node. The depth of the root node is therefore 0. Figure 7 shows nodes at various depths within the tree.

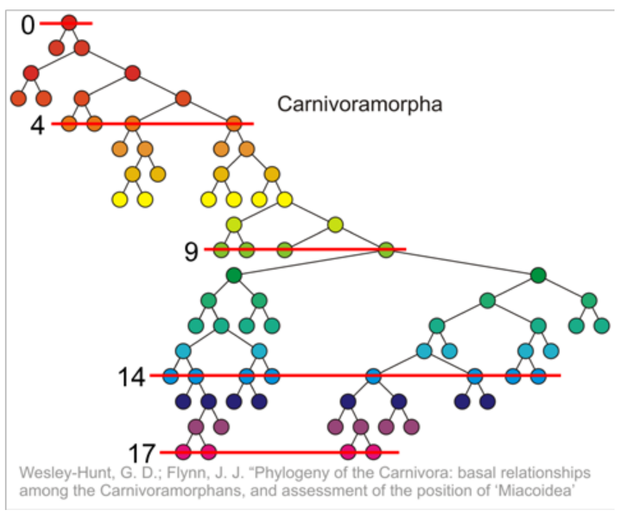

Figure 7. Nodes of various depths.

The *height.of a tree* is the maximum depth of any node within the tree. Consequently, the height of the tree in Figure 7 is 17.

Convention: The height of an empty tree (a tree containing no nodes) is said to be –1.

Definitions: ancestors, strict ancestors, descendants and strict descendants

If a path exists from node *A* to node *B*,

1. *A* is said to be an *ancestor* of *B* and
2. *B* is said to be a *descendant* of *A*.

The root node is therefore an ancestor of all nodes within the tree. If we want to include all descendants of a node except for the node itself, we may use the term *strict descendants* and, similarly, the *strict ancestors* of a node are all ancestors of a node other than the node itself.

Figure 8 shows the strict ancestors of the violet node in red and the strict descendants in blue.

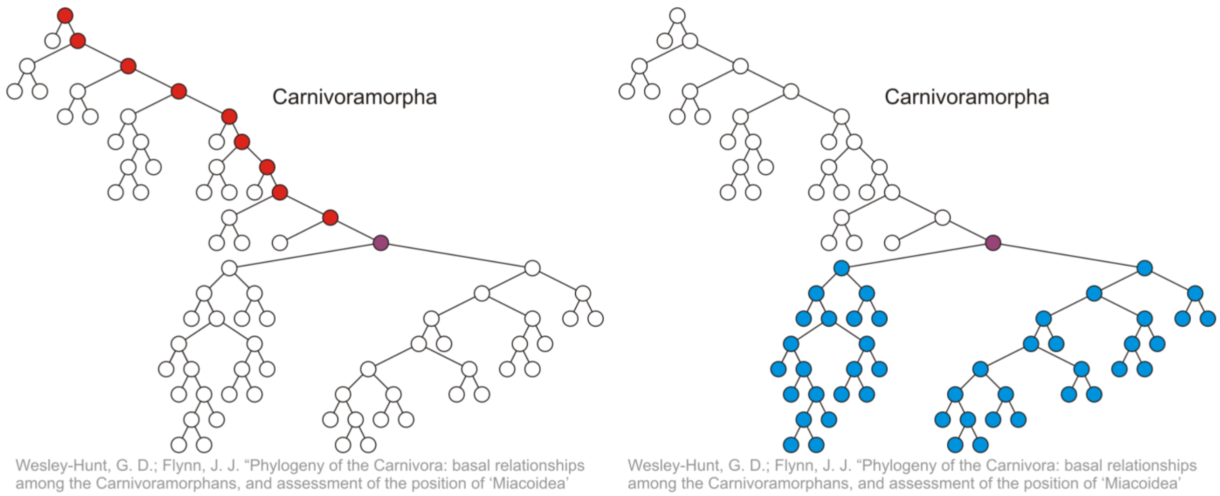

Figure 8. The strict ancestors (red) and strict descendants (blue) of the violet node.

4.1.2 Recursive Definition of a Tree

A recursive definition of a tree is as follows:

1. A degree-0 node is a tree, and
2. A node of degree *n* is a tree if it has *n* children, all of which are trees and none of which have overlapping descendants.

4.1.3 The Tree Structure of XML

The nesting of opening and closing tags within XML defines an ordered tree. For example, the HTML document in Program 1 can be represented by the tree in Figure 9.

Program 1. A simple HTML document.

```
<html>

    <head>
         <title>Hello World!</title>
     </head>
     <body>
         <h1>This is a <u>Heading</u></h1>
         <p>This is a paragraph with some <u>underlined</u> text.</p>
     </body>
</html>
```

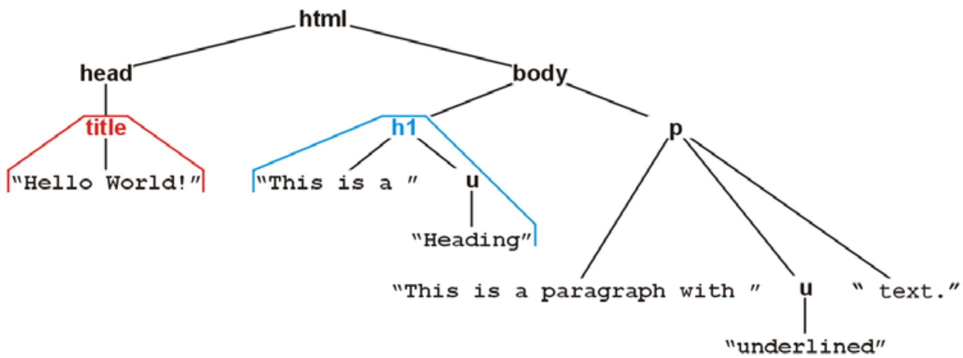

Figure 9. The tree representing the HTML document in Program 1.

All XML documents may be interpreted as ordered rooted trees. This particular document is rendered by a web browser as shown in Figure 10.

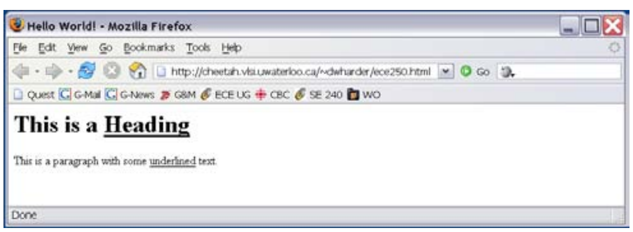

Figure 10. A rendering of the HTML in Program 1.

4.1.3.1 Cascading Style Sheets

Cascading Style Sheets (CSS) manipulate HTML documents by allowing the user to specify decorations on components of the tree. For example, the CSS style

```
<style type="text/css">
    h1 { color:blue; }
</style>
```

states that all visible descendants of an `h1` tag that accept `color` as a decoration should use the colour blue. With this style, the HTML document would be rendered as shown in Figure 11.

Figure 11. The HTML document in Program 1 with the given style.

Now, if you add another entry to the style file:

```
<style type="text/css">
   h1 { color:blue; }
   u { color:red; }
</style>
```

This now says that all descendants of an underlined tag should be coloured red. The issue here is that because the underline tag `<u>` in the title is a sub-tree of the `<h1>` title tag, it is now coloured red, too (the decorations on the sub-tree have precedence), as is shown in Figure 12.

Figure 12. An additional style colouring all underlined text red.

If you only wanted underlined text that is itself a descendant of a paragraph tag `<p>`, you could use the style

```
<style type="text/css">
   h1 { color:blue; }
   p u { color:red; }
</style>
```

where p u specifies that the colour red is to be applied to only underlined text that is itself a descendant of a paragraph tag. This is shown in Figure 13.

Figure 13. Only underlined text within paragraphs is coloured red.

All XML tools make use of this tree representation. Most XML parsers will convert the XML to an internal tree data structure. XML transformation languages can also describe manipulations of the tree format.

As another example, the MathML shown in below may be interpreted by the tree in Figure 14.

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <semantics>
    <mrow><mrow><msup><mi>x</mi><mn>2</mn></msup><mo>+</mo>
    <msup><mi>y</mi><mn>2</mn></msup></mrow>
    <mo>=</mo><msup><mi>z</mi><mn>2</mn></msup></mrow>
    <annotation-xml encoding="MathML-Content">
      <apply><eq/>
        <apply><plus/>
          <apply><power/><ci>x</ci><cn>2</cn></apply>
          <apply><power/><ci>y</ci><cn>2</cn></apply>
        </apply>
        <apply><power/><ci>z</ci><cn>2</cn></apply>
      </apply>
    </annotation-xml>
    <annotation encoding="Maple">x^2+y^2 = z^2</annotation>
  </semantics>
</math>
```

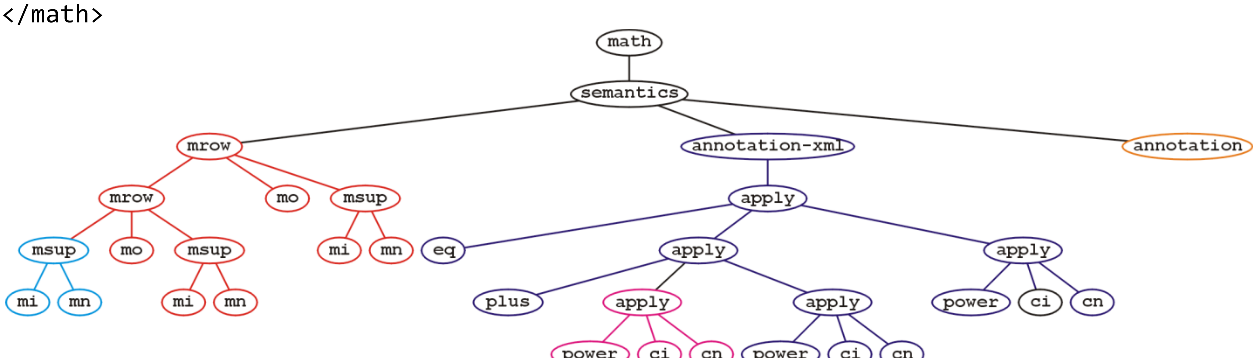

Figure 13. The representing the MathML expression $x^2 + y^2 = z^2$.