<u>***Section 10.5 – Rooted Trees***</u>

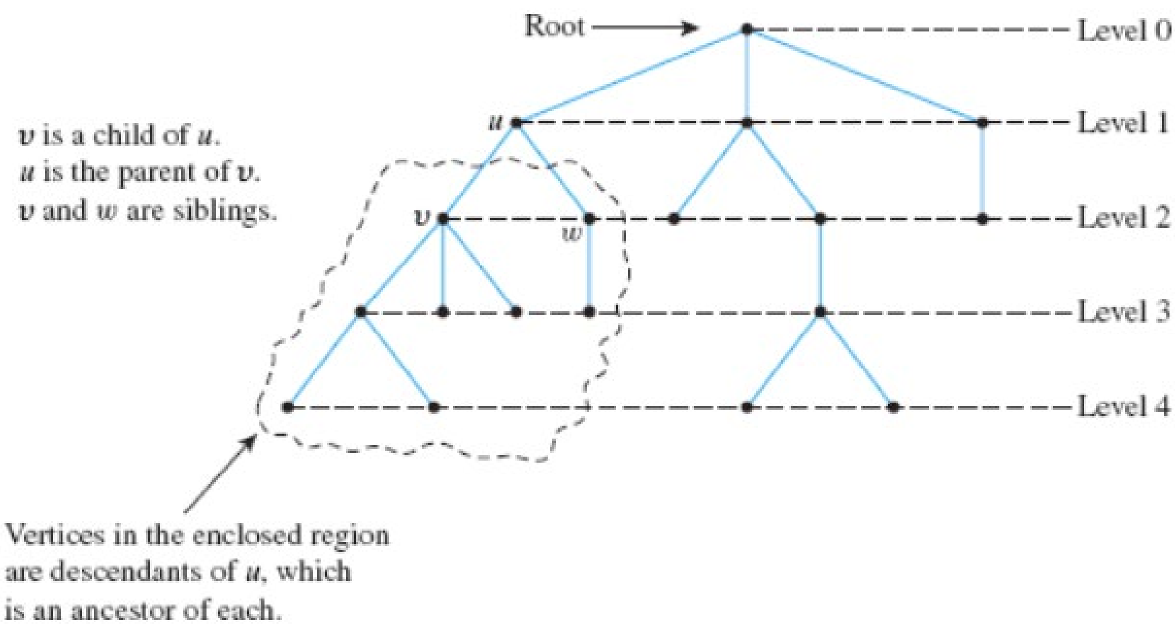
Look outside, how many trees do you see? These trees are rooted.

Some of you may be interested in ancestry and family trees.

Here in our class, the terminology we use for rooted trees blends the language of botanical trees and family trees.

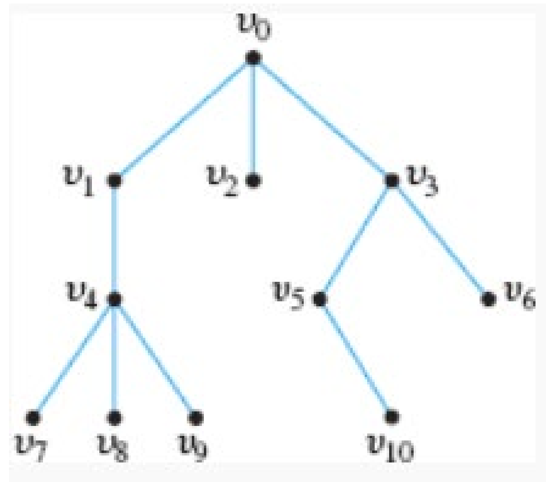
Here we define a ***rooted tree*** as one where one vertex has been distinguished from the others and is called the ***root***.

A **rooted tree** is a tree in which there is one vertex that is distinguished from the others and is called the **root.** The **level** of a vertex is the number of edges along the unique path between it and the root. The **height** of a rooted tree is the maximum level of any vertex of the tree. Given the root or any internal vertex v of a rooted tree, the **children** of v are all those vertices that are adjacent to v and are one level farther away from the root than v. If w is a child of v, then v is called the **parent** of w, and two distinct vertices that are both children of the same parent are called **siblings.** Given two distinct vertices v and w, if v lies on the unique path between w and the root, then v is an **ancestor** of w and w is a **descendant** of v.



Partner up

Consider the tree with root v_0 shown below.



a. What is the level of v_5?

b. What is the level of v_0?

c. What is the height of this rooted tree?

d. What are the children of v_3?

e. What is the parent of v_2?

f. What are the siblings of v_8?

g. What are the descendants of v_3?

h. How many leaves (terminal vertices) are on the tree?

==

Consider the tree with root v_0 shown below.

a. What is the level of v_5? 2

b. What is the level of v_0? 0

c. What is the height of this rooted tree? 3

d. What are the children of v_3? v_5 , v_6

e. What is the parent of v_2? v_0

f. What are the siblings of v_8? v_7 , v_9

g. What are the descendants of v_3? v_5 , v_6 , v_{10}

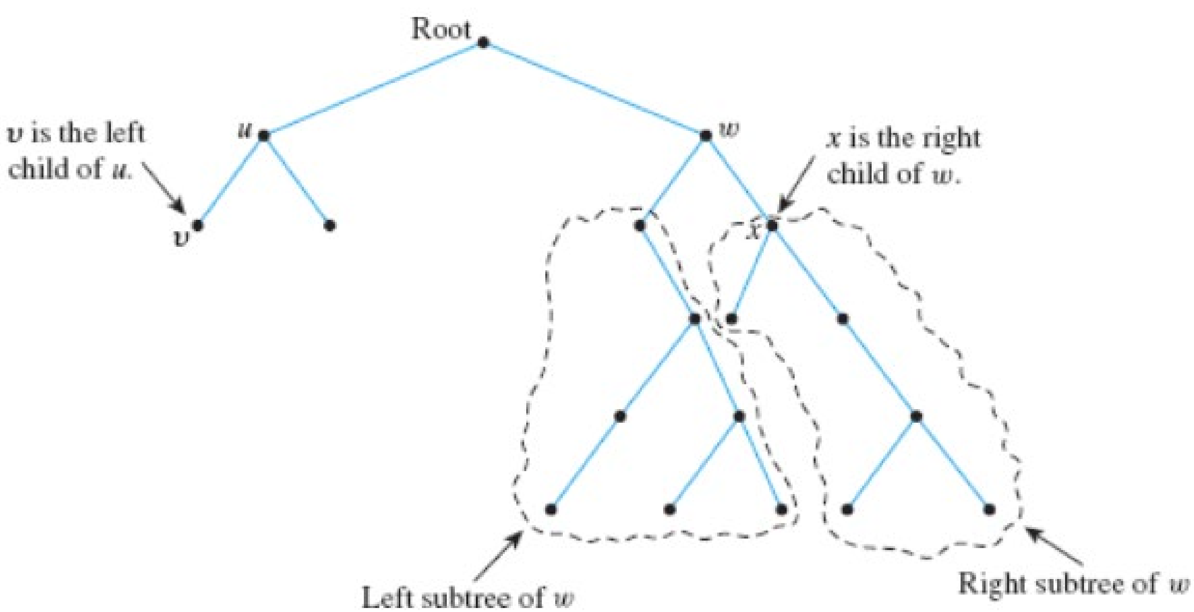
h. How many leaves (terminal vertices) are on the tree? $\boldsymbol{t} = 6 \rightarrow v_2 , v_6 , v_7 , v_8 , v_9 , v_{10}$

Note that in the tree shown below, the root is v_0, v_1 has level 1, v_1 is the child of v_0, and both v_0 and v_1 are leaves (terminal vertices).

A **binary tree** is a rooted tree in which every parent has at most two children. Each child in a binary tree is designated either a **left child** or a **right child** (but not both), and every parent has at most one left child and one right child. A **full binary tree** is a binary tree in which each parent has exactly two children.

Given any parent v in a binary tree T, if v has a left child, then the **left subtree** of v is the binary tree whose root is the left child of v, whose vertices consist of the left child of v and all its descendants, and whose edges consist of all those edges of T that connect the vertices of the left subtree. The **right subtree** of v is defined analogously.

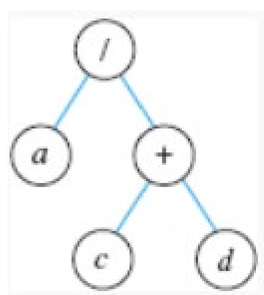
Here's an example Binary Tree with those terms:



We use binary trees to represent expressions, they are a key component in compilers and interpreters.

The internal vertices are the operators, and the terminal vertices/leaves are the operands.

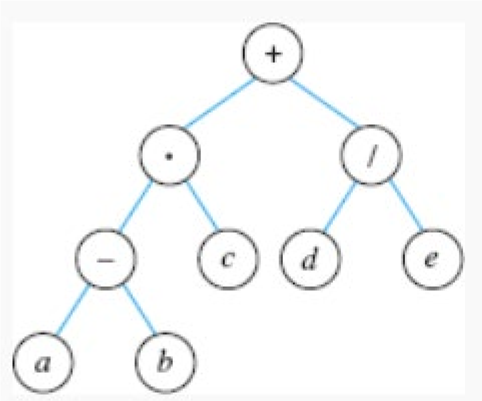
The following tree represents the expression $\frac{a}{c+d}$



Partner up

Draw a binary tree to represent the expression $((a - b) \cdot c) + (d/e)$

==



There is an interesting theorem about binary trees, if you know the number of internal vertices k.

If k is a positive integer and T is a full binary tree with k internal vertices, then

(1) T has a total of $2k+1$ vertices, and

(2) T has $k+1$ leaves.

So, full binary trees have an odd number vertices. If k is an integer, $2k + 1$ is ALWAYS odd.

We can also specify the maximum number of leaves of a binary tree, given its height:

For every integer $h \geq 0$, if T is any binary tree with height h and t leaves, then

$$t \leq 2^h.$$

Equivalently:

$$\log_2 t \leq h.$$

And if our binary tree is not only full, but the leaves are all on the same level, we can change the $\leq$ to $=$:

A full binary tree of height h in which all the leaves have the same level has 2^h leaves.

Is there a full binary tree that has 10 internal vertices and 13 terminal vertices?

No, a full binary tree with 10 internal vertices would have $10 + 1 = 11$ terminal vertices/leaves.

Is there a binary tree that has height 5 and 38 leaves?

No, a height 5 tree would have at most $2^5 = 32$ leaves.