In the course, we cover only rooted trees. In graph theory, a (non-rooted) tree is a minimally connected graph; equivalently, a connected graph with no circuits.

CSI 310: Lecture 21

University at Albany Computer Science Dept.
Each arc expresses the structural relation between the root node and the
subtrees.

An expression (a) Either is an identifier of constant,
or has a top level operator, except:

(1) If it has an operator, it has one
(2) Any operator and operands

or more expressions as operands (no overlap). (and)

The root of each of the trees specified
is common with each other or the root.
Zero or more rooted trees called
its subtrees, with no nodes or arcs in

(1) One root node. (and)
(2) A non-empty tree has:
root. And keep moving upward to each node’s parent, you will eventually reach the
one). Then move again to that node’s parent’s parent (provided there is one).
If you start at any node and move to the node’s parent (provided there is
Each node except for the root has exactly one parent; the root has no parent.

1. There is one special node, called the root.

Each node may be associated with zero or more different nodes, called
its [that node’s children]. If a node c is the child of another node d, then we
say that d is c’s parent. 

A tree is a finite set of nodes. The set might be empty (no nodes), which is
Main/Science definition:

University at Albany Computer Science Dept.
{ 
  if (halt) {
    cout << temp >> " is the root." << endl;
  
  cout >> parent(temp) ? 
  { 
    temp = parent(temp) 
  } 
  while (temp != NOT the root ) 
  
  temp = u;
  node temp;
  } 

find-root(node n) 

The following algorithm, started at any node, always halts, eventually: 

A more formal description of property 4: ...
halts when it is started on \( u \).

The no-cycle property of trees: For every node \( u \), the \( \text{find-root}(u) \) algorithm

\[
\begin{align*}
\{ \\
\text{return } \text{find-root}(\text{parent}(u)) \}; \\
\text{else} \\
\{ \\
\text{return } u; \\
\text{cout } \gg u \gg \text{ is the root. } \gg \text{end}; \\
\text{if } u \text{ is the root } \\
\text{node } \text{find-root}(\text{node } u) \\
\} \\
\end{align*}
\]

Recursive root-finder
The purpose of clause 4. of the definition and not clause 4. relationship which satisfies clauses 1–3 is an example of a parent–child. This is an example of a tree is to exclude situations like nodes 3, 2, 1, 4, 3, etc. where 2 is the parent of 3, 1 is the parent of 2, 4 is the parent of 3, I is the parent of 2, 4 is the unique root is 5.
is c's parent.

left child and its right child. If a node c is the child of node p, we say: "

2 Each node may be associated with up to two other different nodes, called its

Replace M/S tree definition clause (2) by:

left and right

Binary Tree (in brief, each node's children are distinguished from one another as
Complete Binary Tree: Every non-leaf has exactly 2 children.

Full Binary Tree: Every leaf has the same depth.

(William, Cormen, Leiserson, Rivest, and others) call this the height of a tree.

Depth of a leaf: Maximum depth of any of its leaves.

Better authors, writing for more sophisticated readers (like Knuth, Aho, Hopcroft, etc.) introduce new terms:

Depth of a node: Number of parent-to-child steps from the root to this node.

Left and Right subtrees of a node:

Consisting of this new root's descendants.

Subtree (we can view any non-root node as the root of a new, smaller tree).

Parent, Sibling, Ancestor, Descendant.

Computer ScI, examinations and job placement interviews; Tree: A node with no children.

The lingo (technical terminology) the language of computer geeks, tested on
Recursive algorithm to compute the depth of a node:

```java
int depth(Node n) {
    if (n is the root) return 0;

    else return 1 + depth(parent(n));
}
```
Note (2004 correction): You cannot eliminate the if test since height must return 0 when n is a leaf and at least 1 when not.

```cpp
{ }
{ }
return maxex;
{ }
maxex = max(maxex, height(c));
for each of the children nodes c of n
int maxex = 1; // count the step from node n
else if (n is a leaf) return 0;
( int height(node n) }
Recursive algorithm to compute the height of the subtree with the given root:
```
Example of an expression and its Parse Tree

\[ A = (B = ((C - (D \times (E++) + (F \times G))) + F \times G)) \]

Details continued on the next 2 frames.
The top level operator is multiplication (**).
top level operator is increment (++)

* ( ( (D*(E++)))

c - (D*(E++))

D

c - (D*(E++))

E

C

I
The following C++ expression has an expression tree that is binary but not complete binary:

```
(6*(++(XX*)))
```
Binary Search Tree: A decision tree for answering whether or not a given question is in a finite set. By using questions of the form: "Is it $u > v?" and "Is it $u = v?" each node of the tree is in a finite set, by using questions of the form: "Is it $u > v?" and "Is it $u = v?" each node of the tree is in a finite set.

Other Name Space Systems: EC, the Domain Name System of the Internet.

File Name Trees: Express a system to identify files using a sequence of search for "human".


Expression Trees: Express the structure of the computation expressed by an application.
For the tree and each subtree \( L \), the root contains the largest of the numbers in the tree (of numbers) with the heap property.

Heap ordered tree ("Heap"): A tree (of numbers) with the heap property.
Worst case: use an amount of computer work proportional to the height of the tree, in the insertion into a heap, and search and insertion in a binary search tree.

Main conclusion: number into a heap ordered tree.

We then viewed Main/Savitch's page 520 and described how to insert a new search tree.

example of how a dictionary of states (of the USA) is implemented by a binary search and insertion a new number into it.

We viewed Main's slide set 10b on the dictionary abstract data type and the search and search and insertion how to insert a decision tree.

We viewed Main/Savitch's page 498 on a binary search tree and described how to insert a decision tree.