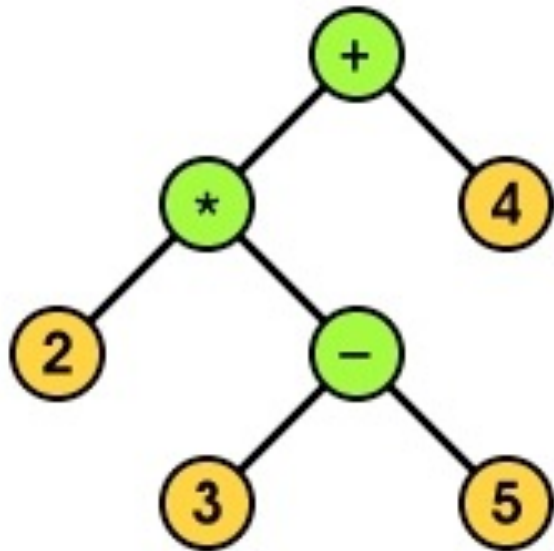


Announcements

MP4 available, due 10/14, 11:59p.

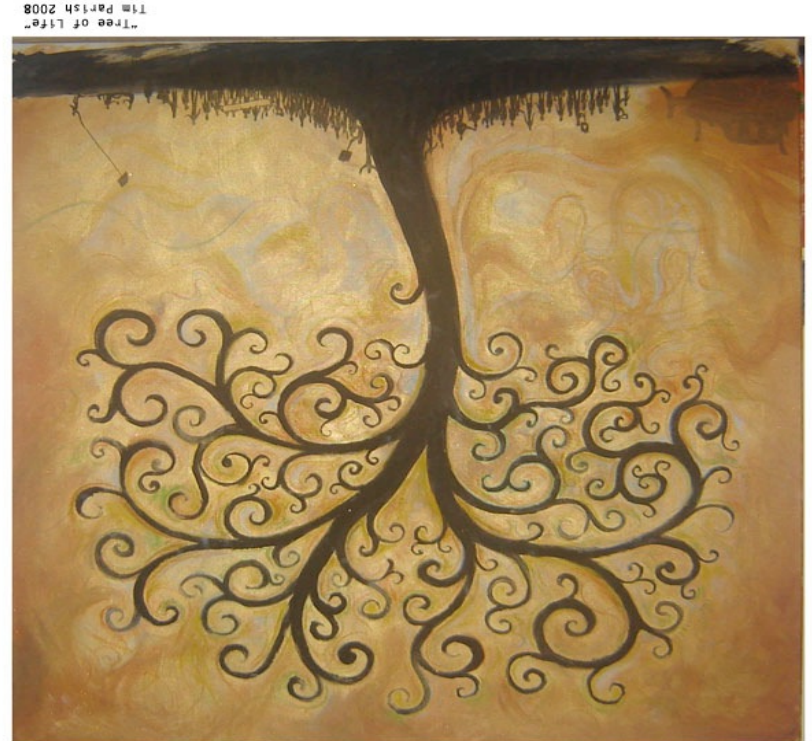
Exam3: 10/16-10/19.

Binary tree, recursive definition:



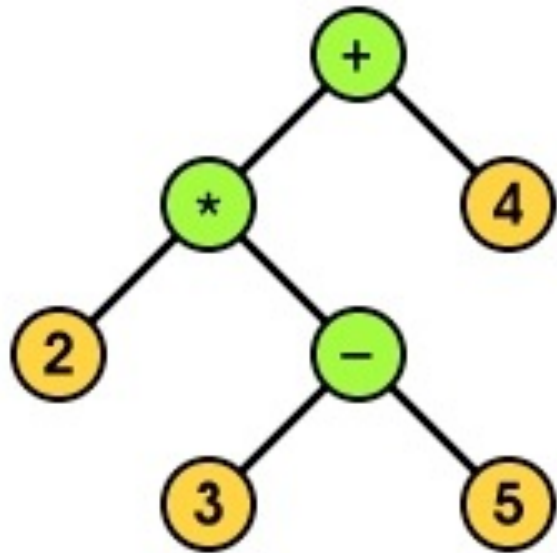
A binary tree T is either

-
- OR
-



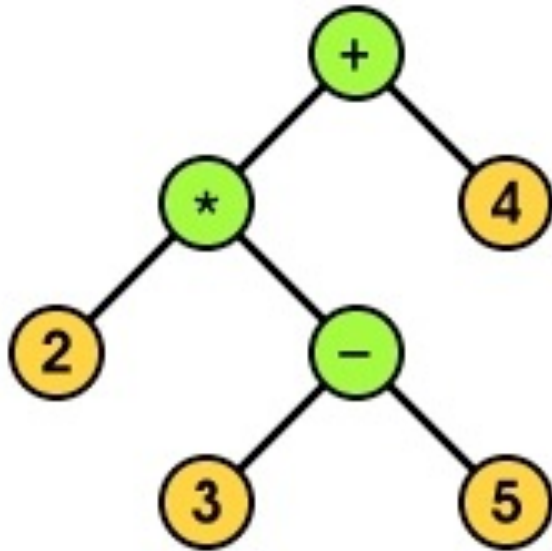
An (important) example of a function on a binary tree:

$\text{height}(t)$ -- length of longest path from root to a leaf



Given a tree T , write a recursive defn of the height of T , $\text{height}(T)$:

Full Binary tree: a tree in which every node has 2 or 0 children



F is a full binary tree if and only if:

- $F = \{\}$ OR,
- $F = \{r, T_L, T_R\}$, and

Perfect Binary tree:

Perfect tree of height h , P_h :

- P_{-1} is an empty tree
- if $h > -1$, then P_h is $\{r, T_L, T_R\}$,
where T_L and T_R are P_{h-1} .

P_0 :

P_2 :

P_1 :

Check for understanding:

How many nodes in a perfect tree of height h ?

Complete Binary tree: for any level k in $[0, h-1]$, level k has 2^k nodes, and on level h , all nodes are "pushed to the left."

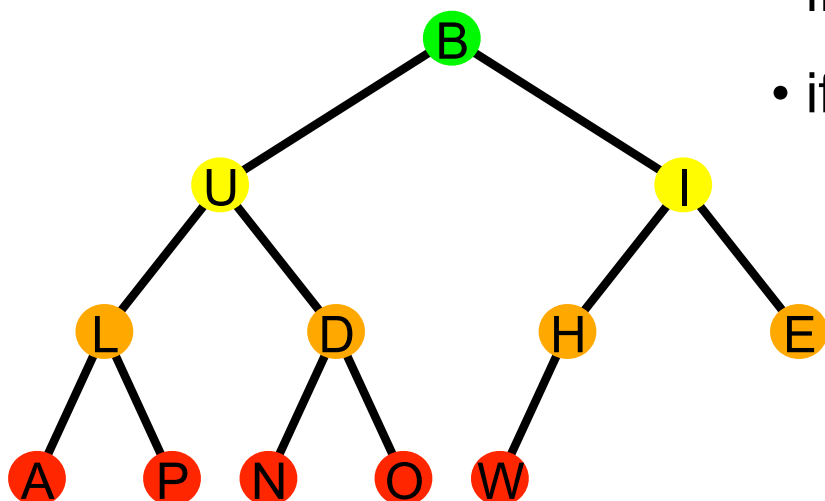
Complete tree of height h , C_h :

- if $h = -1$, then C_h is $\{\}$
- if $h > -1$, then C_h is $\{r, T_L, T_R\}$, and either:

T_L is _____ and T_R is _____

OR

T_L is _____ and T_R is _____



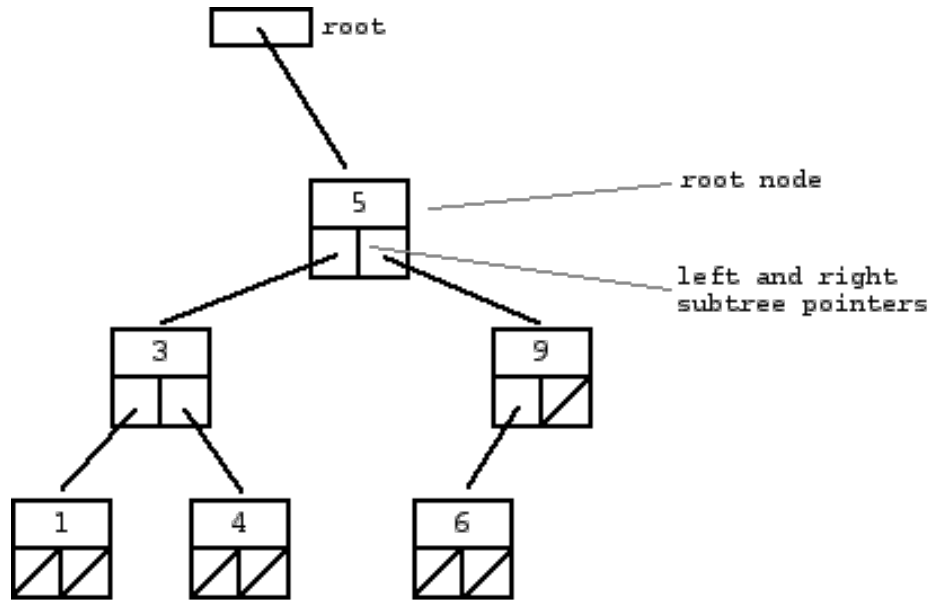
<http://xlinux.nist.gov/dads//HTML/completeBinaryTree.html>

Check for understanding:

Is every full tree complete?

Is every complete tree full?

Rooted, directed, ordered, binary trees



Tree ADT:

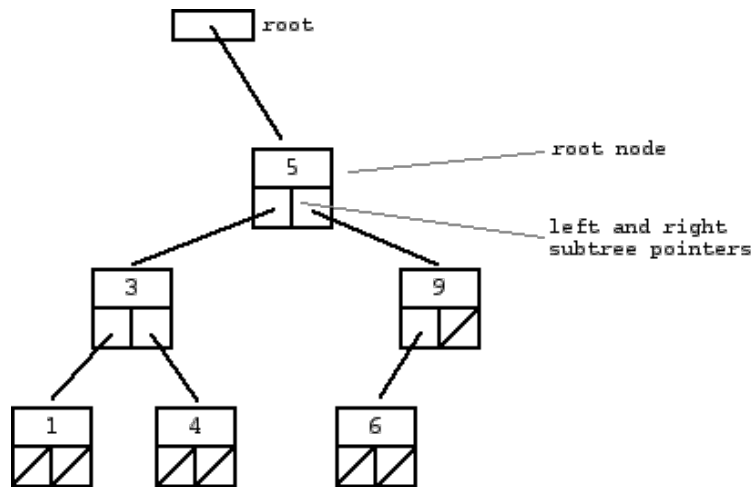
insert

remove

traverse

```
template <class T>
class tree{
public:
...
private:
    struct treeNode{
        T data;
        treeNode * left;
        treeNode * right;
    };
    treeNode * root
...
};
```

Theorem: if there are n data items in a binary tree, then there are _____ null pointers.



Traversal running time...

