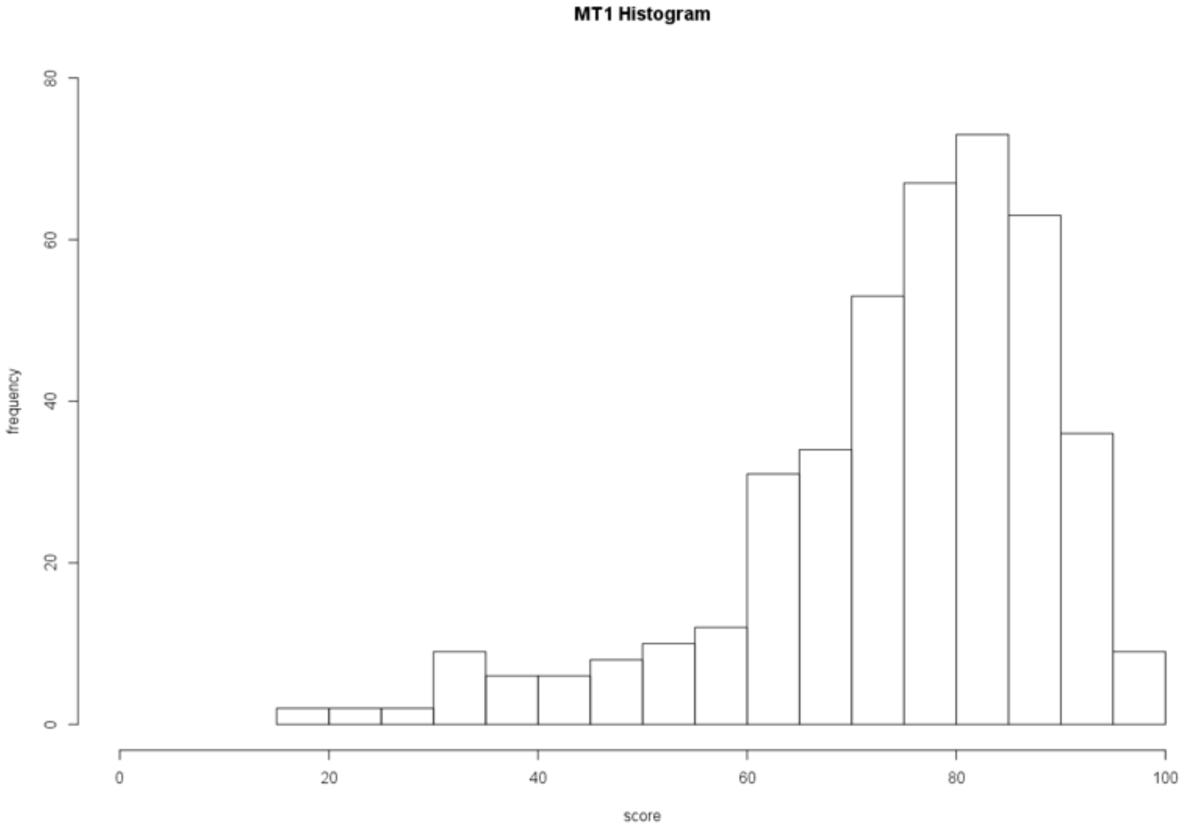
Announcements

MP4 available, due 3/8, 11:59p. EC due 3/1, 11:59p.

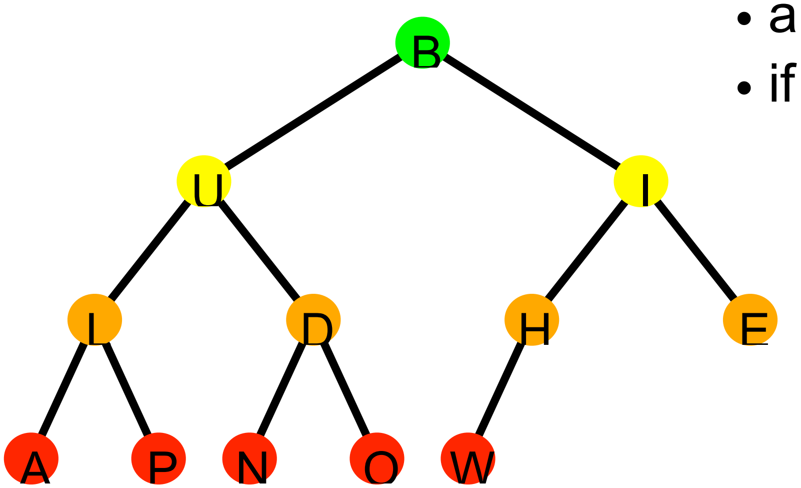
Code Challenge #1: Winners!

Exam Visitation: 3/4, 7:30p, Siebel 0216

TODAY: tree definitions
traversals



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:

- an empty tree is C_{-1}
- 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 _________

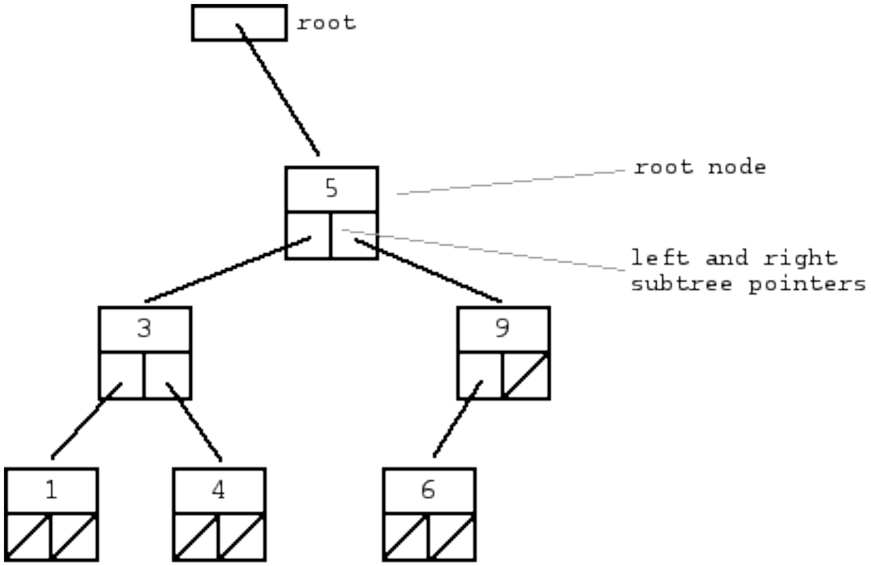
http://xlinux.nist.gov/dads//HTML/completeBinaryTree.html

Check for understanding:

Is every full tree complete?

Is every complete tree full?

Rooted, directed, ordered, binary trees

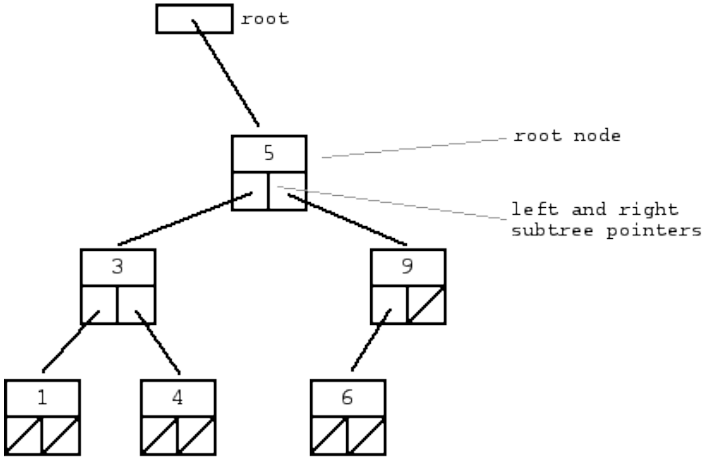


```
template <class T>
class tree{
public:
…
private:
   struct treeNode{
      T data;
      treeNode * left;
      treeNode * right;
   };
   treeNode * root
…
};
```

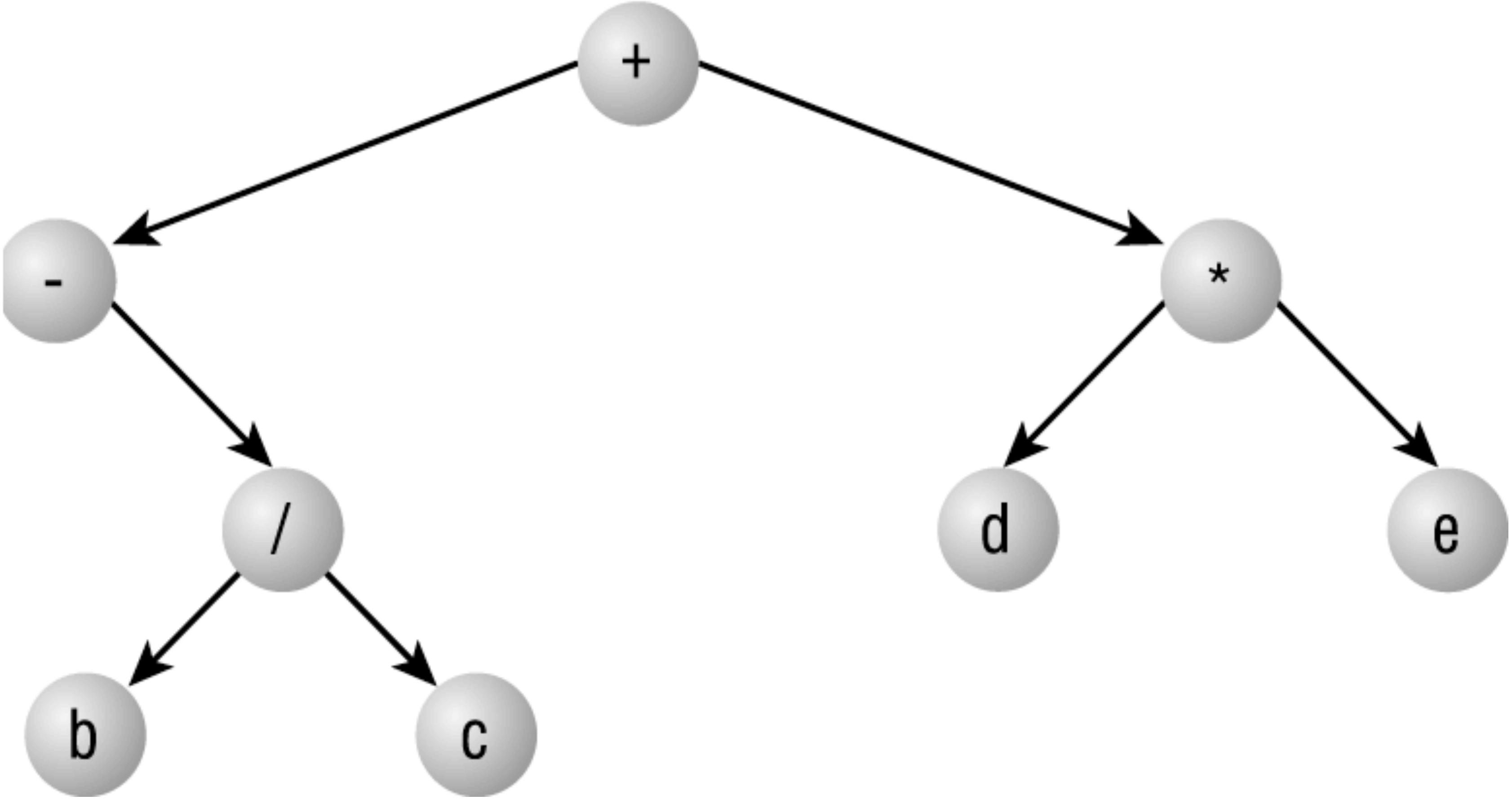
Tree ADT:

- insert
- remove
- traverse

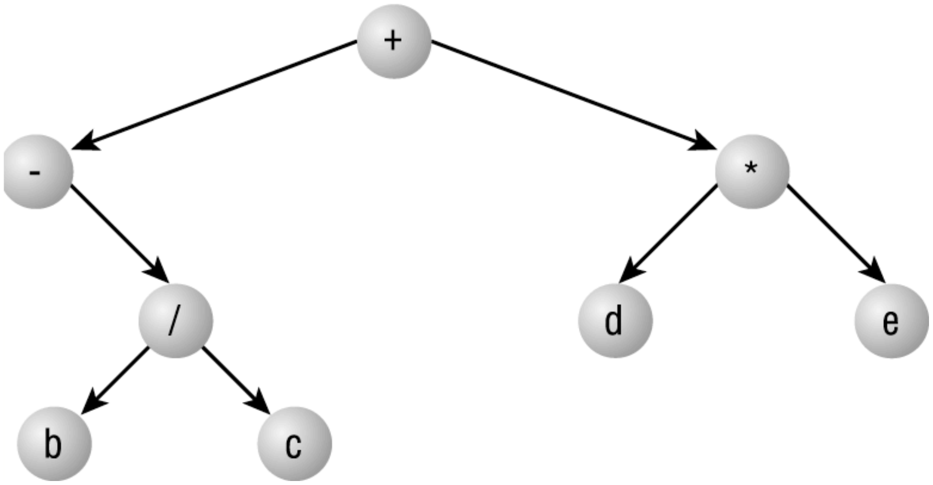
Theorem: if there are n data items in a binary tree, then there are ______ null pointers.



Traversal – scheme for visiting every node.



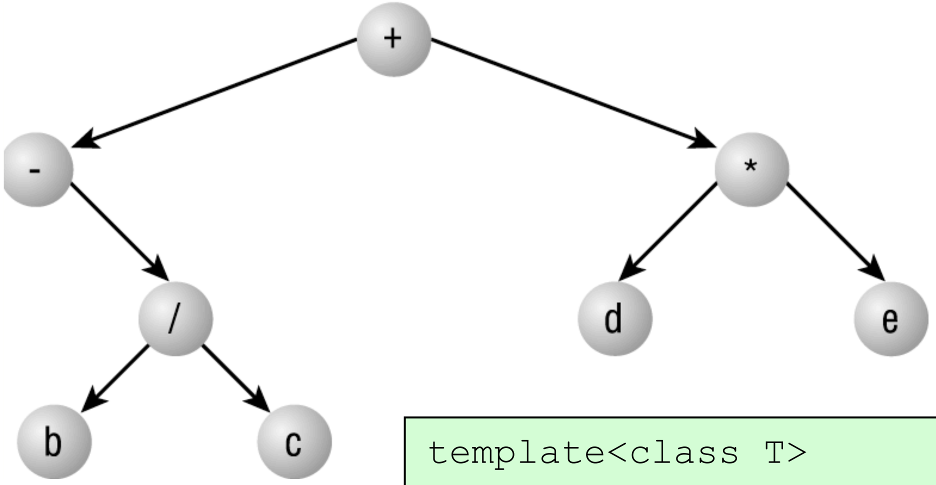
Traversal – scheme for visiting every node.



- At each node, two choices for direction (left, right)
- After both subtrees of a node are complete, move back up tree
- Each node is “visited” 3 times in a traversal.
- Each of those visit times corresponds to a particular kind of traversal.

http://nova.umuc.edu/~jarc/idsv/lesson1.html

Traversals:



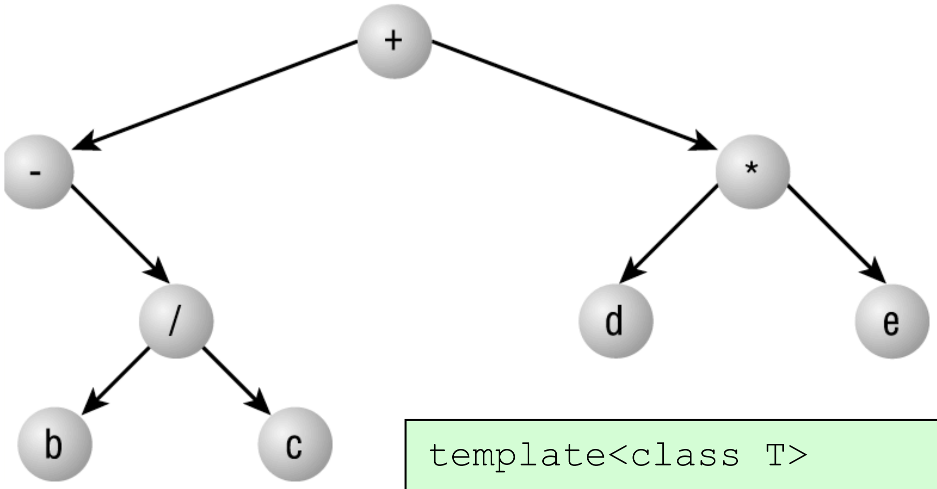
```
template<class T>
void binaryTree<T>::_______Order(treeNode * croot){
    if (croot != null){


        ________Order(croot->left);


        ________Order(croot->right);


    }
}
```

Traversals:



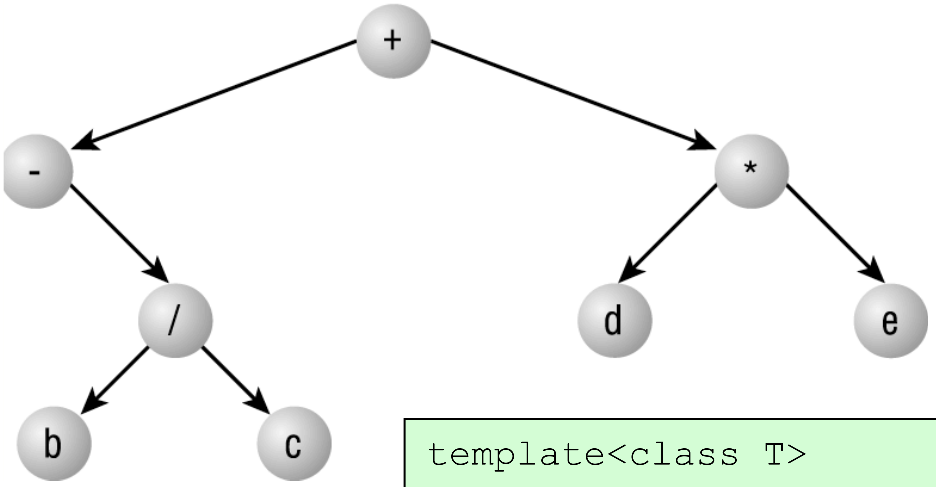
```
template<class T>
void binaryTree<T>::_______Order(treeNode * croot){
    if (croot != null){

         ________Order(croot->left);

         ________Order(croot->right);

    }
}
```

Traversals:



```
template<class T>
void binaryTree<T>::_______Order(treeNode * croot){
    if (croot != null){

         ________Order(croot->left);

         ________Order(croot->right);


    }
}
```

Traversals: A few mechanical questions…

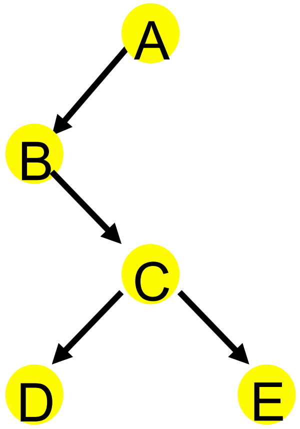
What is the 2nd letter printed in an inOrder traversal of this tree?

In what position is C printed in a postOrder traversal of this tree?

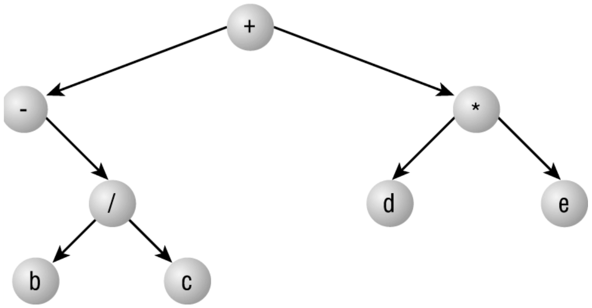
Which traversal prints the data of the tree in ABC order?

Draw and label a tree containing 8 integers so that an inOrder traversal of the tree prints the numbers in order.

Draw and label a tree containing 8 integers so that a preOrder traversal of the tree prints the numbers in order.



Traversals: A few discussion questions…



```
template<class T>
void binaryTree<T>::preOrder(treeNode * croot){
    if (croot != null){
         yell(croot->data);
         preOrder(croot->left);
         preOrder(croot->right);
    }
}
```

What is running time?

Is preOrder public or private?

How could we make this function employ a different function upon a visit?