

Binary Search Trees: Insert



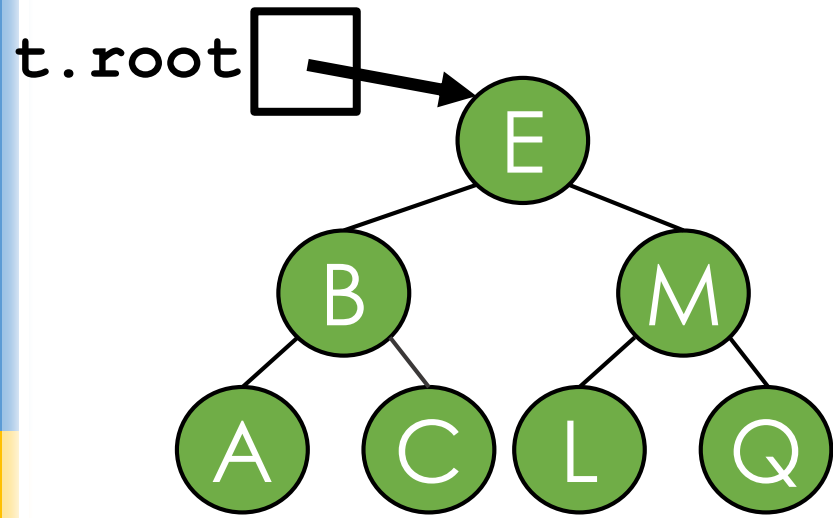
Let's write it!



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

Looking ahead...

- You will implement insert in a different kind of tree in the project, but the ideas are the same

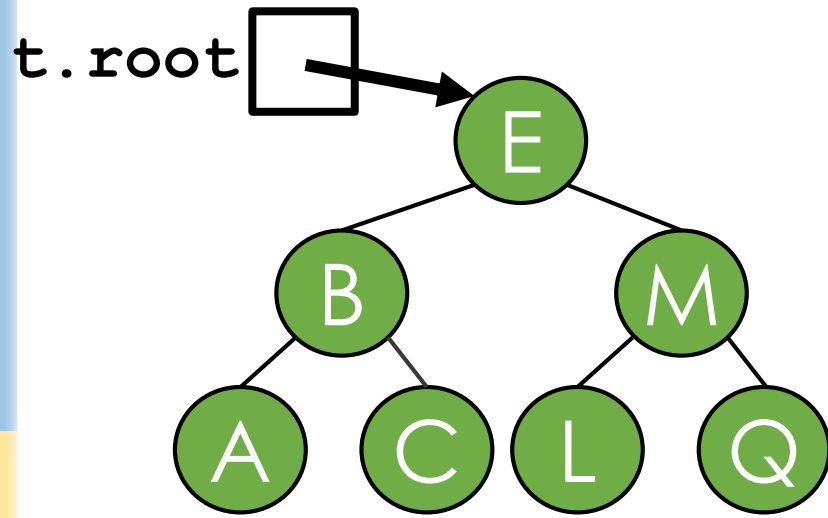


```
t.insert('D')
```

```
public boolean insert(E toInsert)
{
```

find toInsert

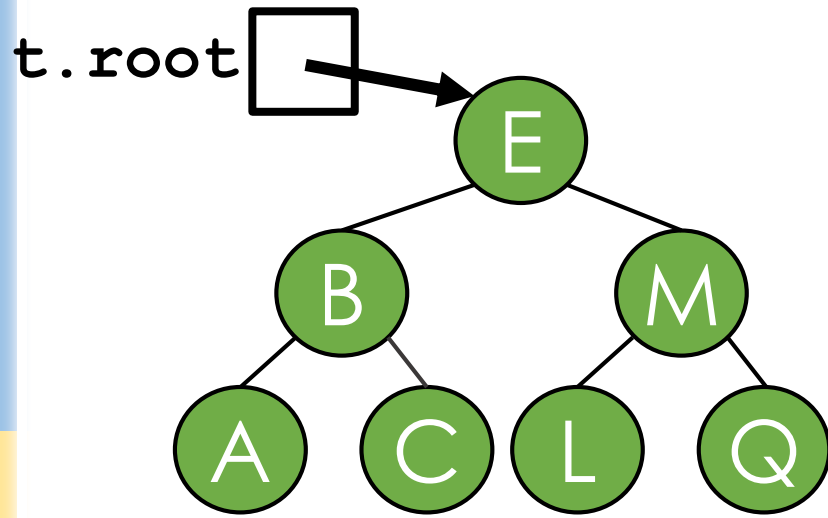
**if it's not there, add it
where you end up**



`t.insert('D')`

```
public boolean insert(E toInsert)
{
    TreeNode<E> curr = root;
    int comp =
        toInsert.compareTo(curr.getData());
    while (curr != null) {
        if (comp < 0)
            curr = curr.getLeft();
        else if (comp > 0)
            curr = curr.getRight();
        else
            return false;
    }
}
```

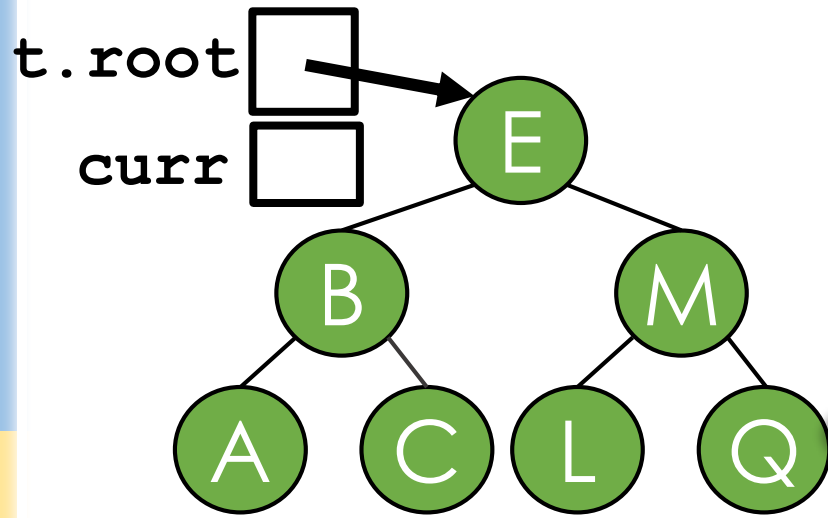
**if it's not there, add it
where you end up**



`t.insert('D')`

```
public boolean insert(E toInsert)
{
    TreeNode<E> curr = root;
    int comp =
        toInsert.compareTo(curr.getData());
    while (curr != null) {
        if (comp < 0)
            curr = curr.getLeft();
        else if (comp > 0)
            curr = curr.getRight();
        else
            return false;
    }
    curr = new TreeNode<E>(toInsert, curr);
}
```

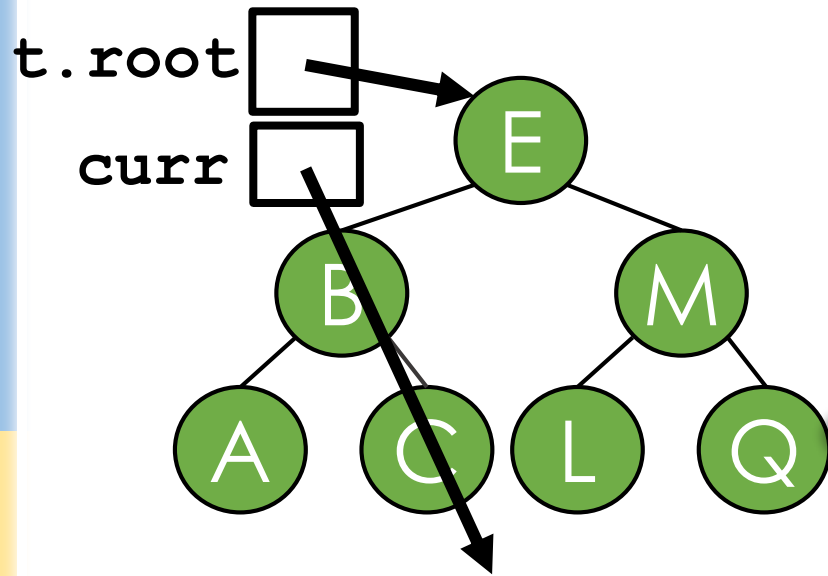
Does this work?



`t.insert('D')`

```
public boolean insert(E toInsert)
{
    TreeNode<E> curr = root;
    int comp =
        toInsert.compareTo(curr.getData());
    while (curr != null) {
        if (comp < 0)
            curr = curr.getLeft();
        else if (comp > 0)
            curr = curr.getRight();
        else
            return false;
    }
    curr = new TreeNode<E>(toInsert, curr);
}
```

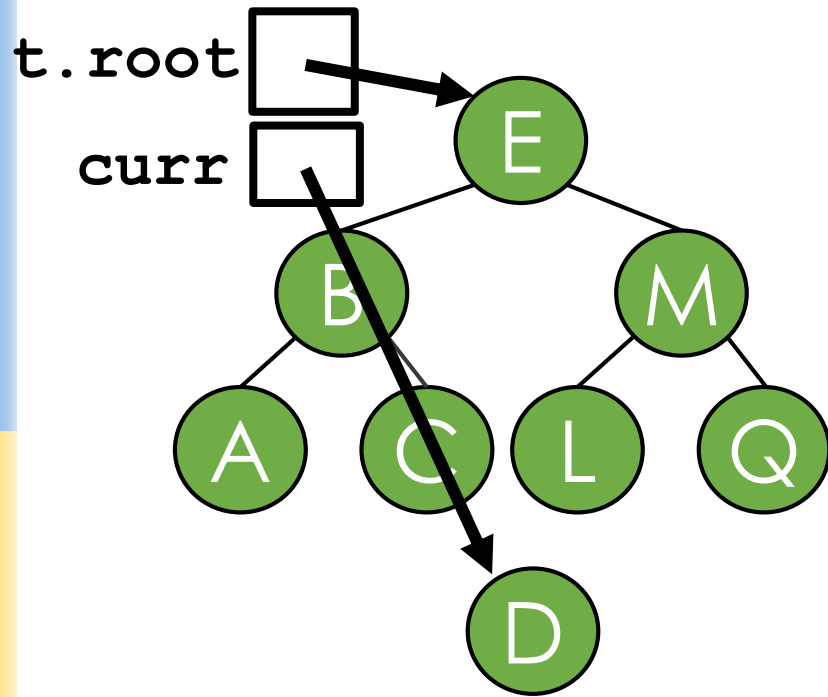
Does this work? NO!



`t.insert('D')`

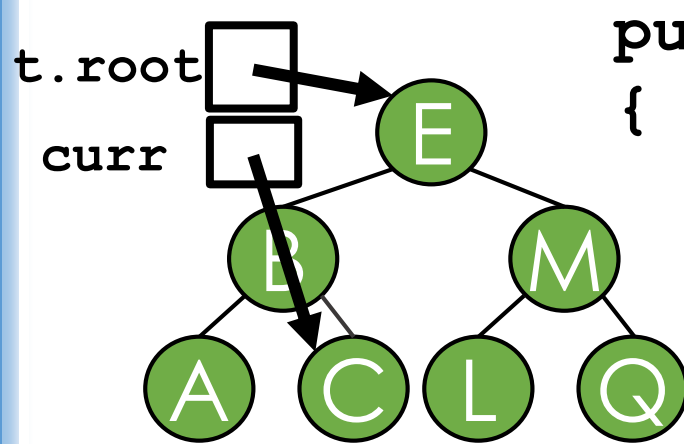
```
public boolean insert(E toInsert)
{
    TreeNode<E> curr = root;
    int comp =
        toInsert.compareTo(curr.getData());
    while (curr != null) {
        if (comp < 0)
            curr = curr.getLeft();
        else if (comp > 0)
            curr = curr.getRight();
        else
            return false;
    }
    curr = new TreeNode<E>(toInsert, curr);
}
```

Does this work? NO!



```
public boolean insert(E toInsert)
{
    TreeNode<E> curr = root;
    int comp =
        toInsert.compareTo(curr.getData());
    while (curr != null) {
        if (comp < 0)
            curr = curr.getLeft();
        else if (comp > 0)
            curr = curr.getRight();
        else
            return false;
    }
    curr = new TreeNode<E>(toInsert, curr);
}
```

Does this work? NO!



```
public boolean insert(E toInsert)
```

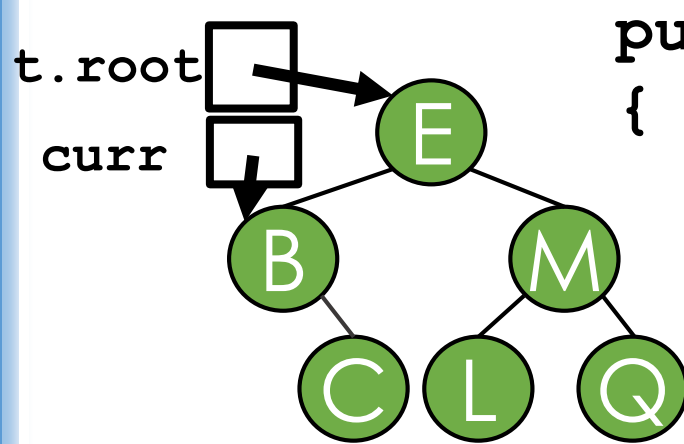
```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
t.insert('D')
```

**Need to stop before
we fall off the end**



```
public boolean insert(E toInsert)
```

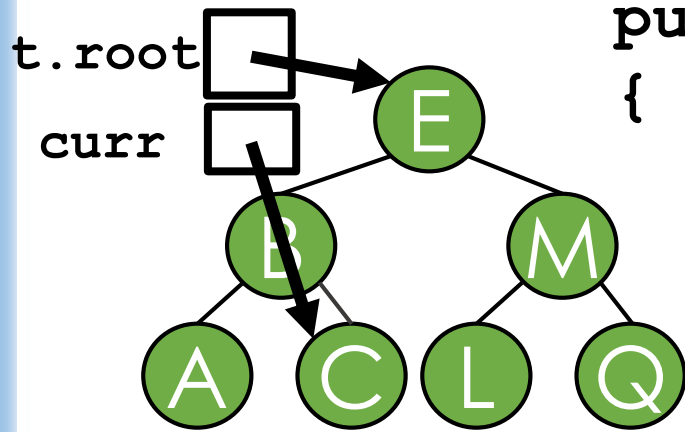
```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
t.insert('D')
```

**Need to stop before
we fall off the end**



```
public boolean insert(E toInsert)
{
```

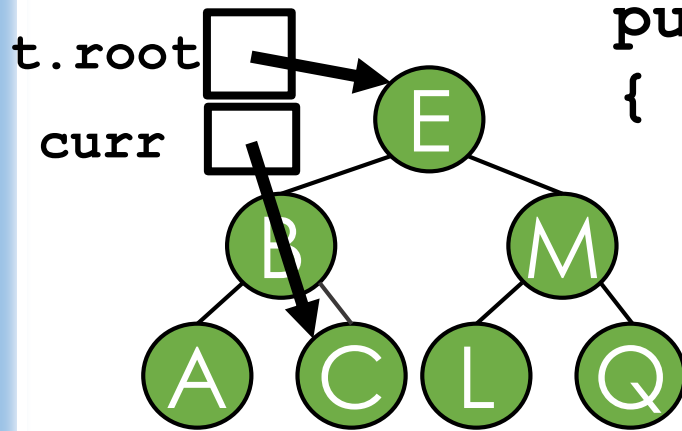
```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||
           comp > 0 && curr.getRight() != null)
```

```
t.insert('D')
```

**Need to stop before
we fall off the end**



```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||  
           comp > 0 && curr.getRight() != null)
```

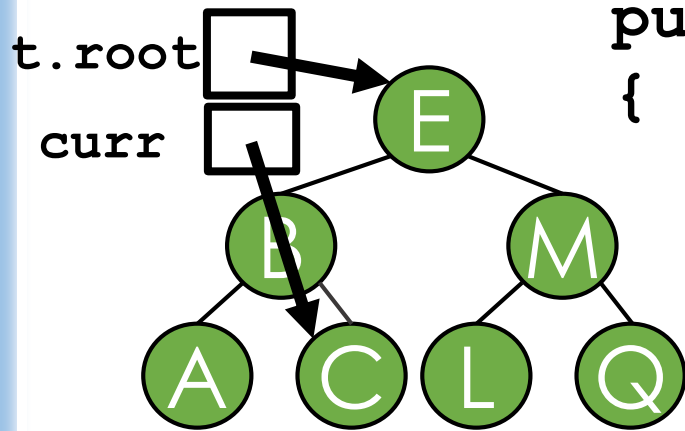
```
{
```

```
    if (comp < 0) curr = curr.getLeft();
```

```
    else curr = curr.getRight();
```

```
t.insert('D')
```

**Need to stop before
we fall off the end**



```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||
           comp > 0 && curr.getRight() != null)
```

```
{
```

```
    if (comp < 0) curr = curr.getLeft();
```

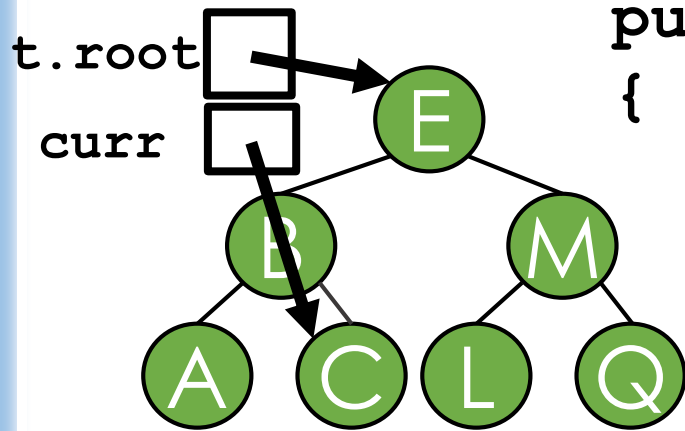
```
    else curr = curr.getRight();
```

```
    comp = toInsert.compareTo(curr.getData());
```

```
t.insert('D')
```

```
}
```

What's next?



```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||
           comp > 0 && curr.getRight() != null)
```

```
{
```

```
    if (comp < 0) curr = curr.getLeft();
```

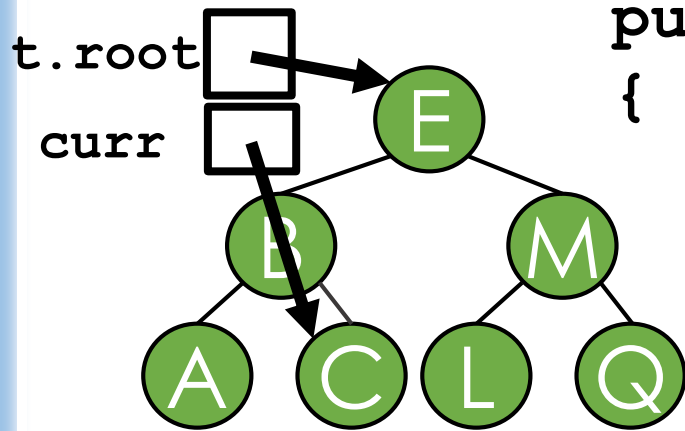
```
    else curr = curr.getRight();
```

```
    comp = toInsert.compareTo(curr.getData());
```

```
t.insert('D')
```

```
}
```

After the loop either: (1) curr points to the last node,
or (2) we found the element



`t.insert('D')`

```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||
           comp > 0 && curr.getRight() != null)
```

```
    {
```

```
        if (comp < 0) curr = curr.getLeft();
```

```
        else curr = curr.getRight();
```

```
        comp = toInsert.compareTo(curr.getData());
```

```
    }
```

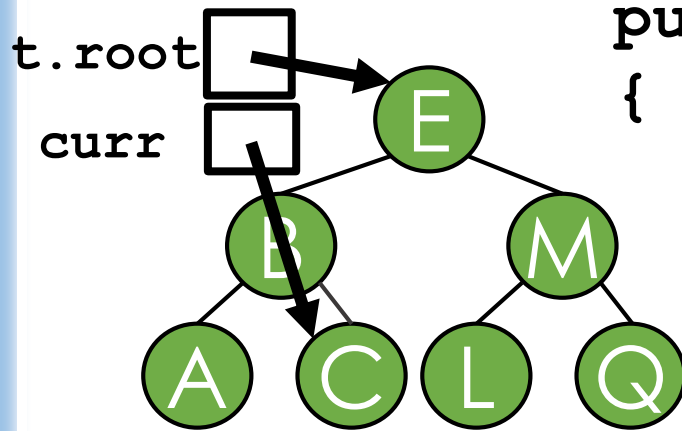
```
    if (comp < 0)
```

```
        curr.setLeftChild(toInsert, curr);
```

```
    else if (comp > 0)
```

```
        curr.setRightChild(toInsert, curr);
```

After the loop either: (1) curr points to the last node,



`t.insert('D')`

```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||  
           comp > 0 && curr.getRight() != null)
```

```
    {
```

```
        if (comp < 0) curr = curr.getLeft();
```

```
        else curr = curr.getRight();
```

```
        comp = toInsert.compareTo(curr.getData());
```

```
    }
```

```
    if (comp < 0)
```

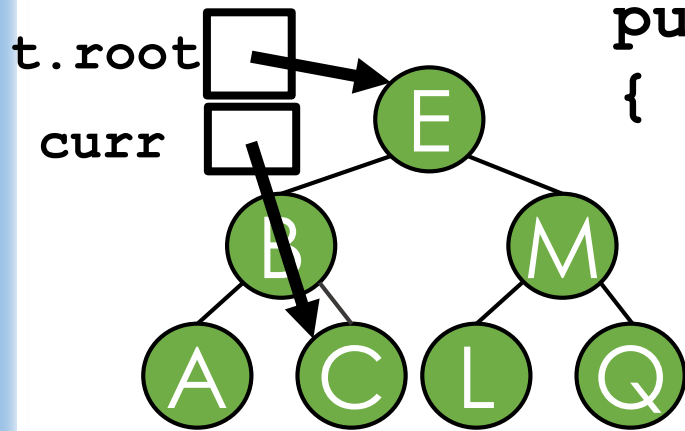
```
        curr.setLeftChild(toInsert, curr);
```

```
    else if (comp > 0)
```

```
        curr.setRightChild(toInsert, curr);
```

```
    else return false;
```

After the loop either: (1) curr points to the last node,
or (2) we found the element



`t.insert('D')`

```
public boolean insert(E toInsert)
```

```
{
```

```
    TreeNode<E> curr = root;
```

```
    int comp = toInsert.compareTo(curr.getData());
```

```
    while (comp < 0 && curr.getLeft() != null ||  
           comp > 0 && curr.getRight() != null)
```

```
    {
```

```
        if (comp < 0) curr = curr.getLeft();
```

```
        else curr = curr.getRight();
```

```
        comp = toInsert.compareTo(curr.getData());
```

```
    }
```

```
    if (comp < 0)
```

```
        curr.setLeftChild(toInsert, curr);
```

```
    else if (comp > 0)
```

```
        curr.setRightChild(toInsert, curr);
```

```
    else return false;
```

```
    return true;
```

```
}
```