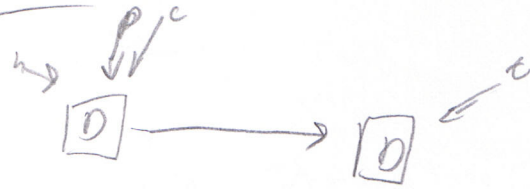


Cap 338



①
10.8.09

to get ID of tail

```
Node head = new Node(0, new Node(0, null));
```

```
return head.next.next = null;
```

Removing current

```
previous.next = new Node(x, previous.next);
```

public class LinkedList

```
{  
    Node tail = new Node(0, null);  
    Node head = new Node(0, tail);  
}
```

```
Node current = head;  
Node previous = head;
```

public boolean isEmpty

```
{  
    return head.next == null;  
}
```

public void insertBefore(int x)

```
{  
    current = new Node(x, previous.next);  
    previous.next = current;  
}
```

public void insertAfter(int x)

```
{  
    previous = previous.next;  
    insertBefore(x);  
}
```

Cmp 338

2

10.8.09

Binary Tree

```
public class NodeBT
```

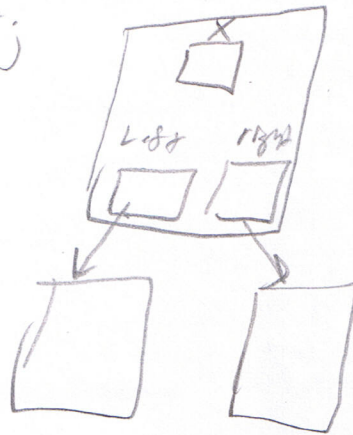
```
{
```

```
    int x
```

```
    // some data
```

```
    NodeBT left;
```

```
    NodeBT right;
```

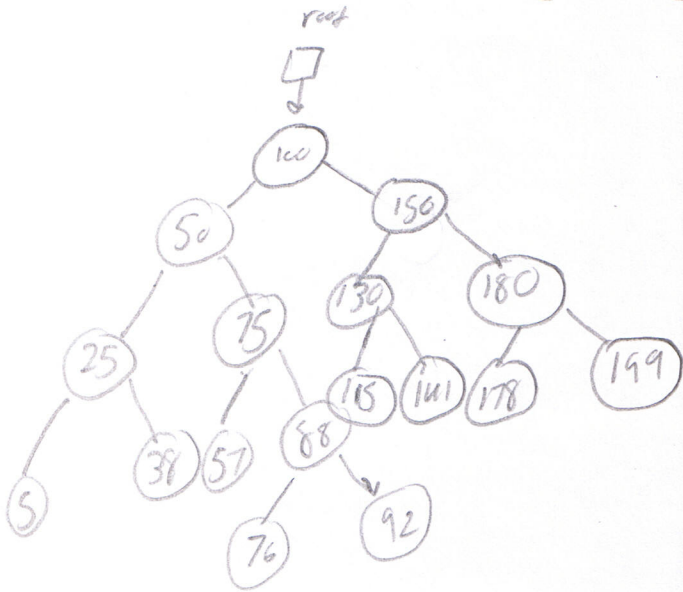


Log to
Cut a
number
in half

I like
to cut!

Def:

Binary tree
on Test



Cnp 335

3

10.8.07

Def: Binary Tree

← will
Be on
test

Empty
(no nodes)

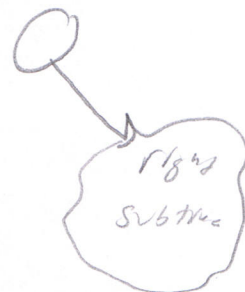
Root node
only



Root node
& a Left
Subtree

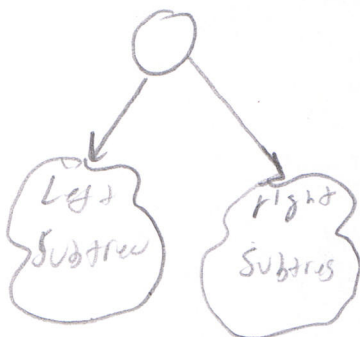


Root node &
a right subtree



Binary tree

Root node &
Left & right
Subtrees

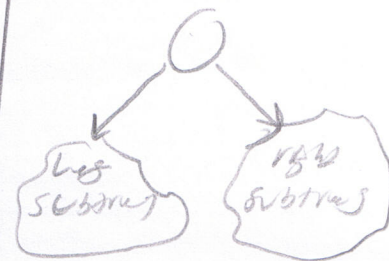


Empty
(no nodes)

Root node
only



Root node &
Left & right
Subtrees



Where each subtree is a binary tree.

Cmp 338

(4)

10.8.09

Def: Binary Search tree

- 1) A Binary Tree
 - 2) where each node in the left subtree is less than the root node.
 - 3) where each node in the right subtree is greater than the root node
 - 4) Every subtree is a binary search tree.
-

Ex: Infix Expr

expr that express