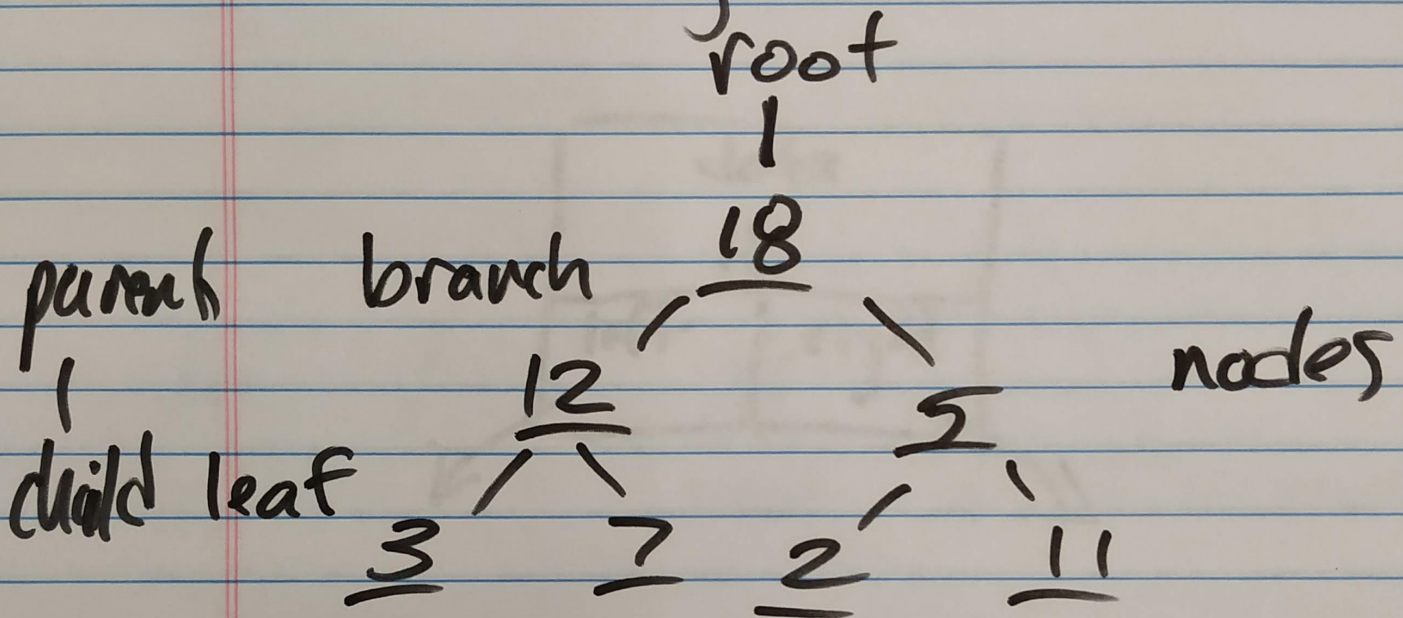
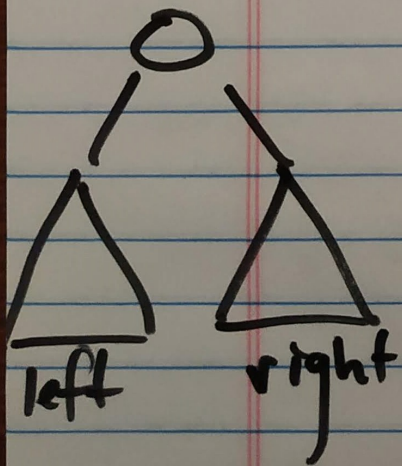


# Binary Trees

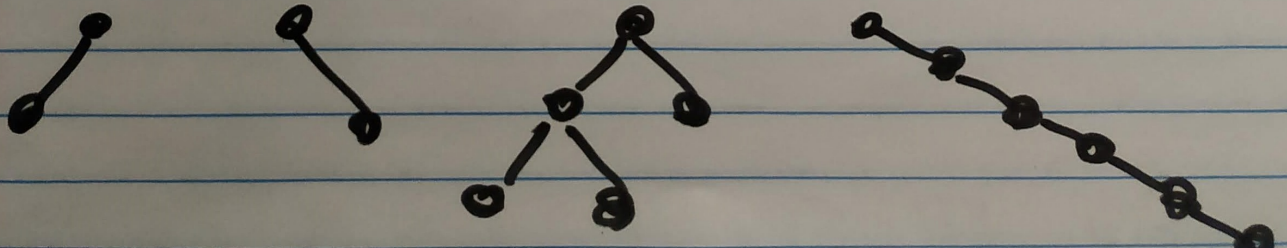


A binary tree is either:

- an empty tree
- a root node with left and right subtrees

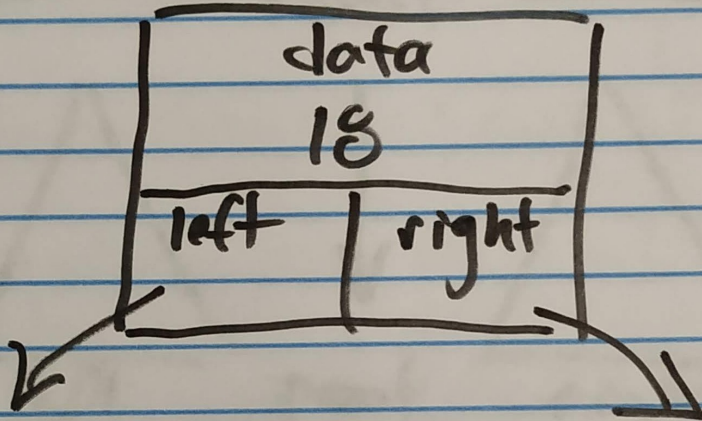


(empty)





# IntTree Node



```
IntTreeNode root = new IntTreeNode(18);  
root.left = ...  
root.right = ...
```

```
public class IntTreeNode {
    public int data;
    public IntTreeNode left;
    public IntTreeNode right;

    public IntTreeNode(int data) {
        this(data, null, null);
    }

    public IntTreeNode(int data, IntTreeNode left,
                       IntTreeNode right) {
        this.data = data;
        this.left = left;
        this.right = right;
    }
}
```

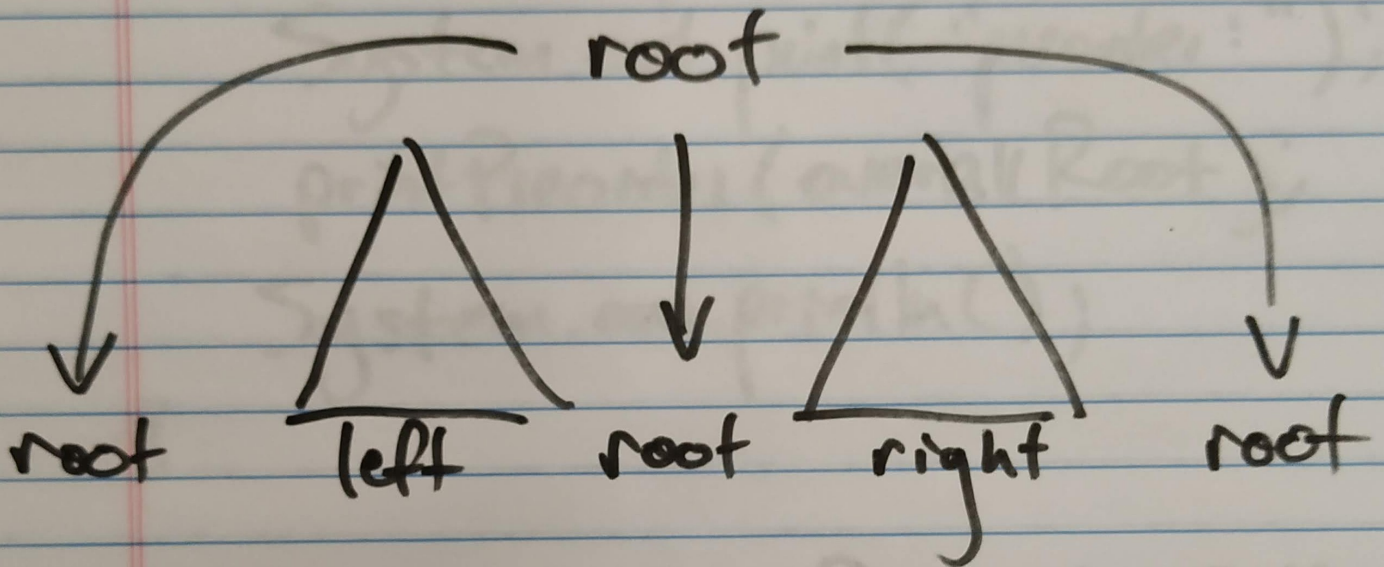
```
public class IntTree {
    private IntTreeNode overallRoot;
```

<Methods>

3



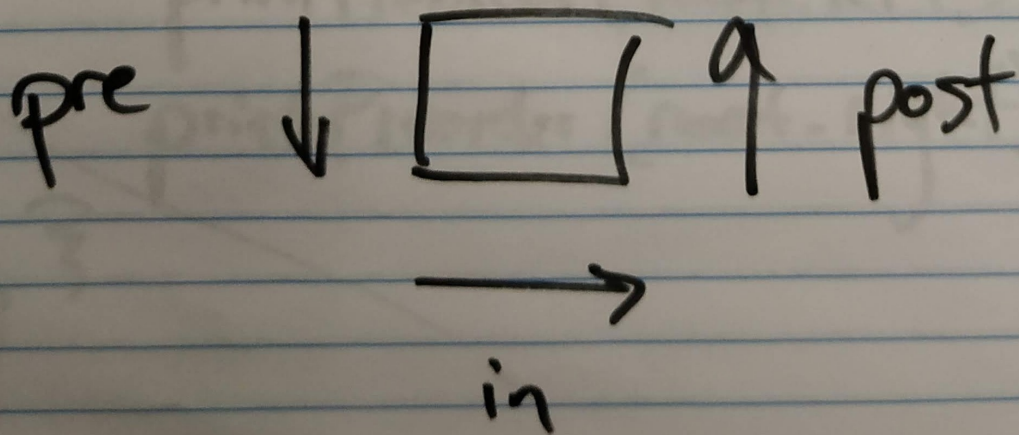
# Tree Traversal

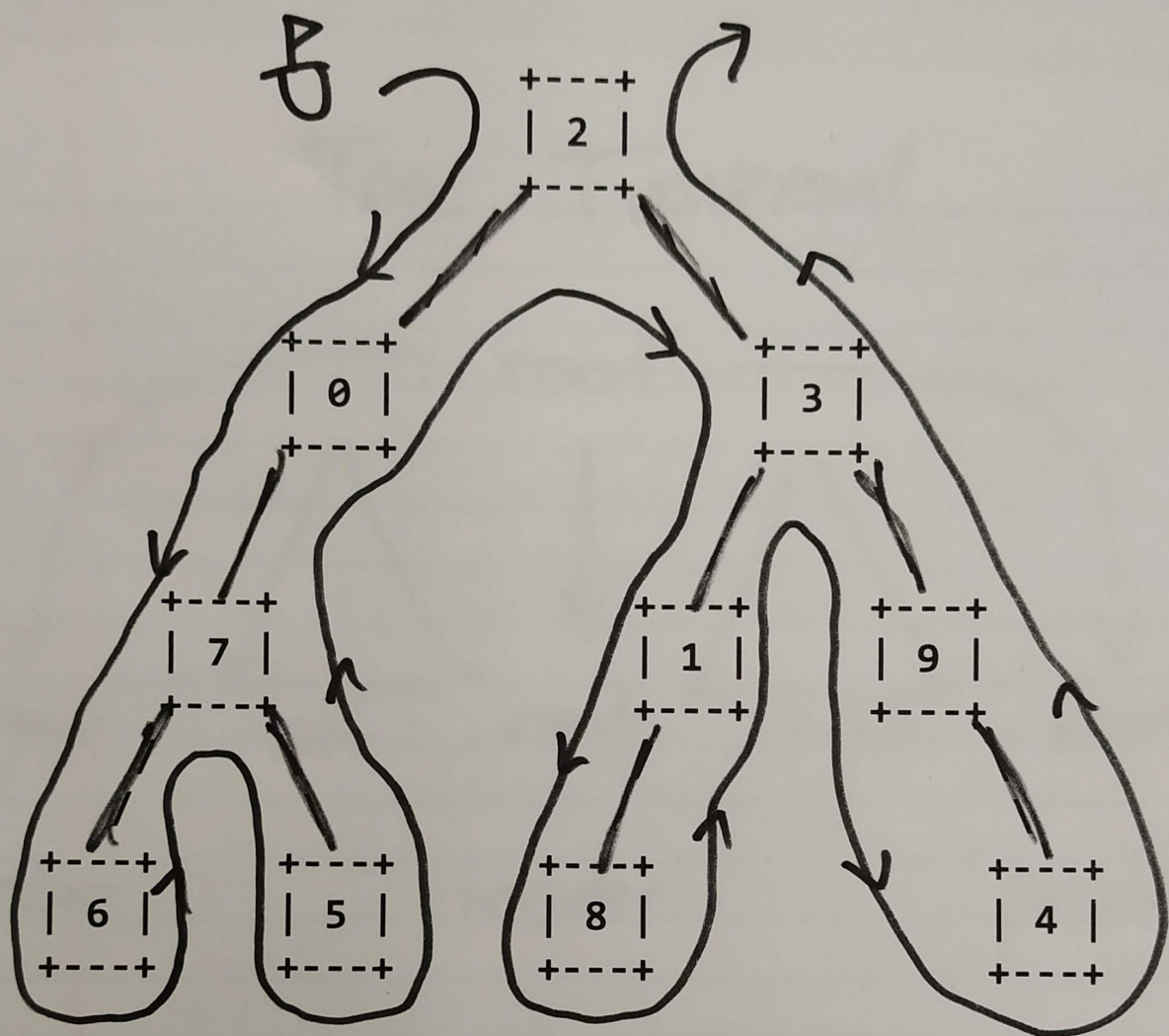


preorder

inorder

postorder





preorder: 2 0 7 6 5 3 1 8 9 4

inorder: 6 7 5 0 2 8 1 3 9 4

postorder: 6 5 7 0 8 1 4 9 3 2



```
public void printPreorder() {  
    System.out.print("preorder:");  
    printPreorder(overallRoot);  
    System.out.println();  
}
```

```
private void printPreorder(InTreeNode root) {  
    if (root != null) {  
        System.out.print(" " + root.data);
```

```
        printPreorder(root.left);  
        printPreorder(root.right);
```

```
    }  
}
```