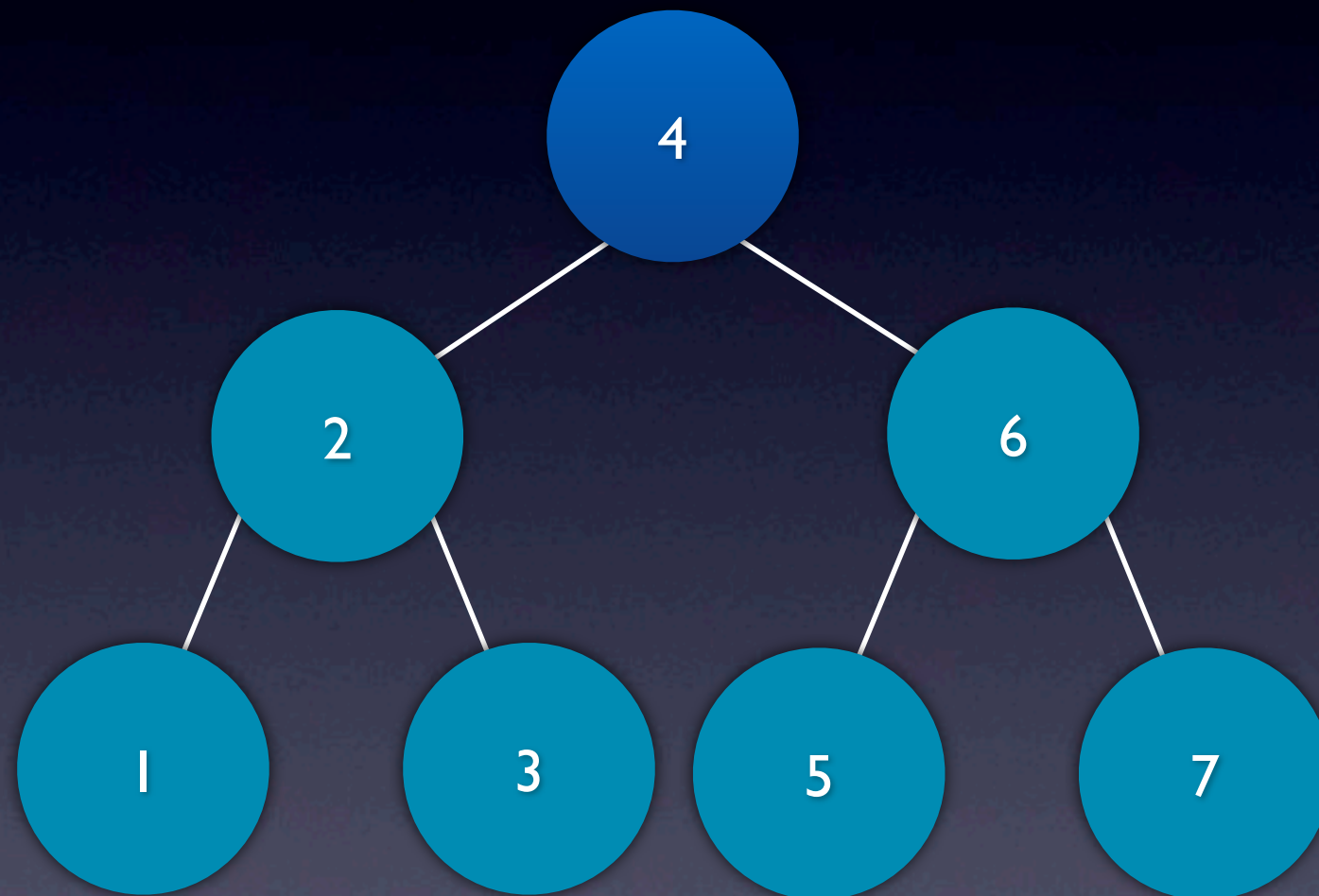


Binärbäume

Aufbau

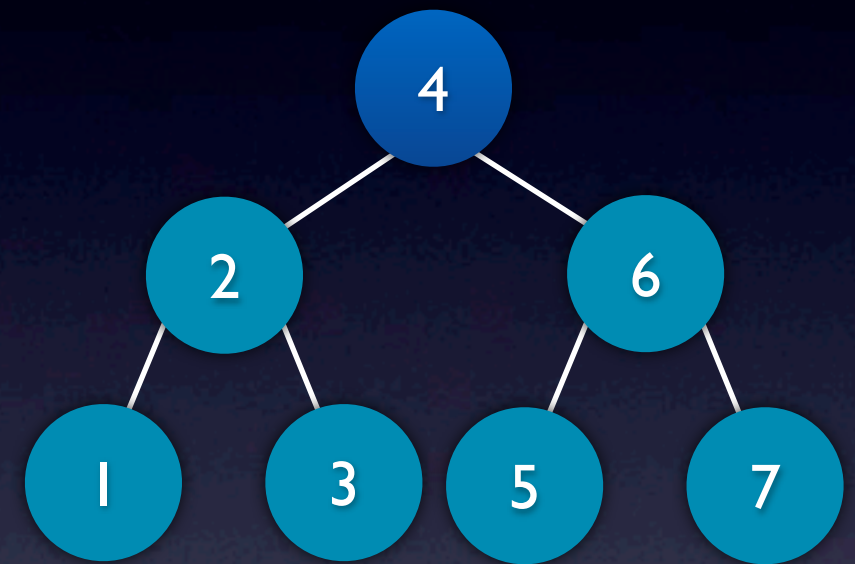


Implementation

```
public abstract class TreeNode {  
    protected TreeNode left;  
    protected TreeNode right;
```

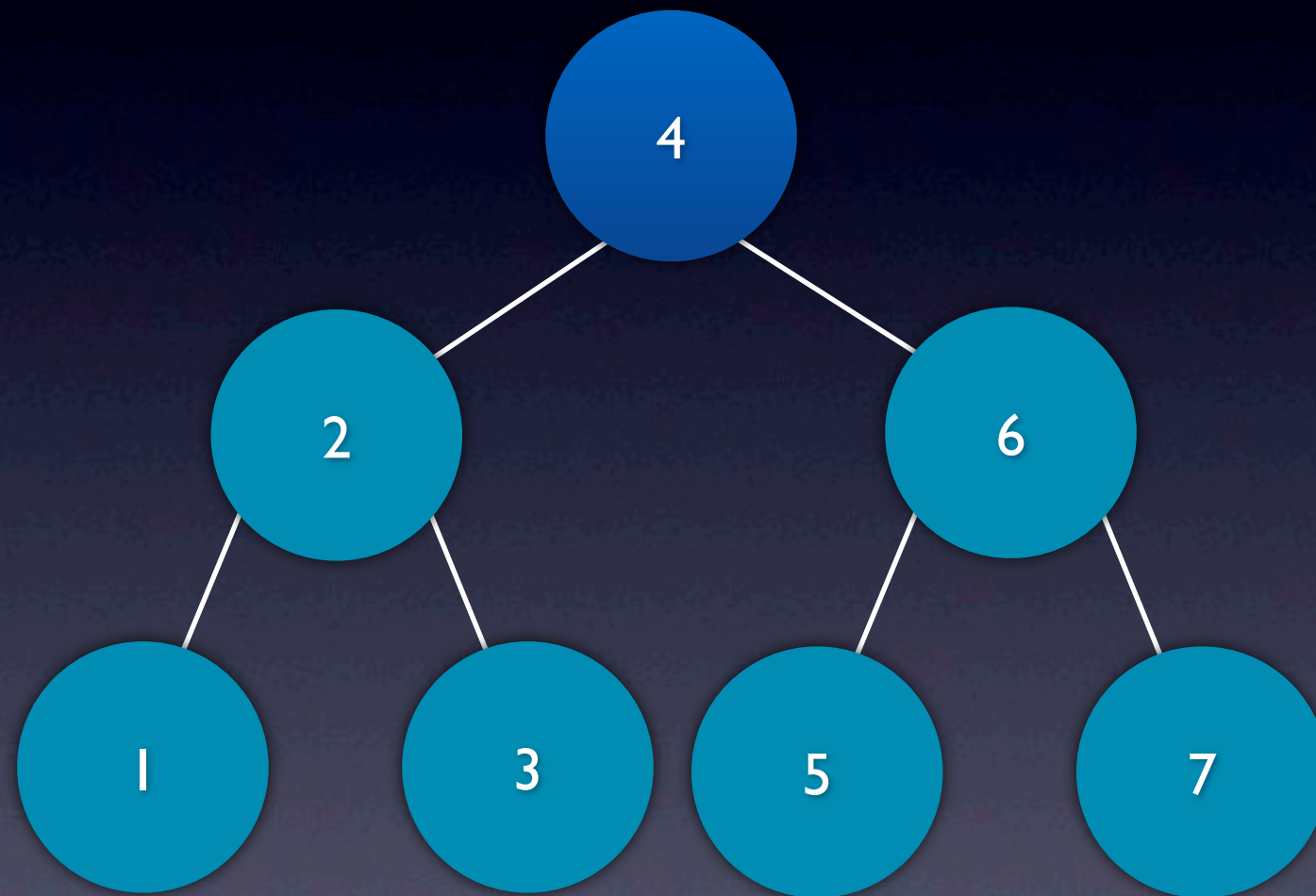
```
    public TreeNode() {  
        this(null, null);  
    }
```

```
    public TreeNode(TreeNode left, TreeNode right) {  
        this.left = left;  
        this.right = right;  
    }  
}
```



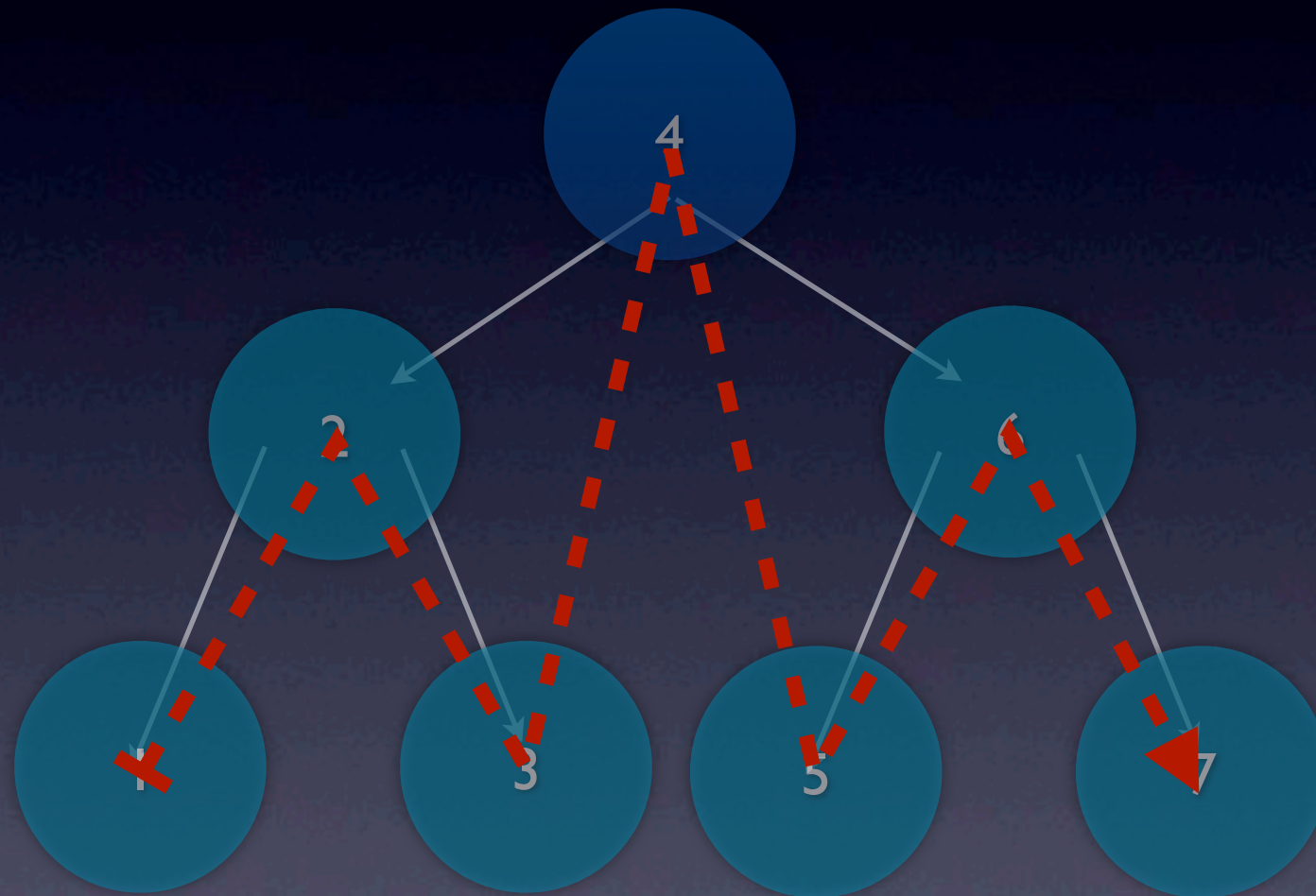
(evtl. key und value)

Durchlaufen: InOrder



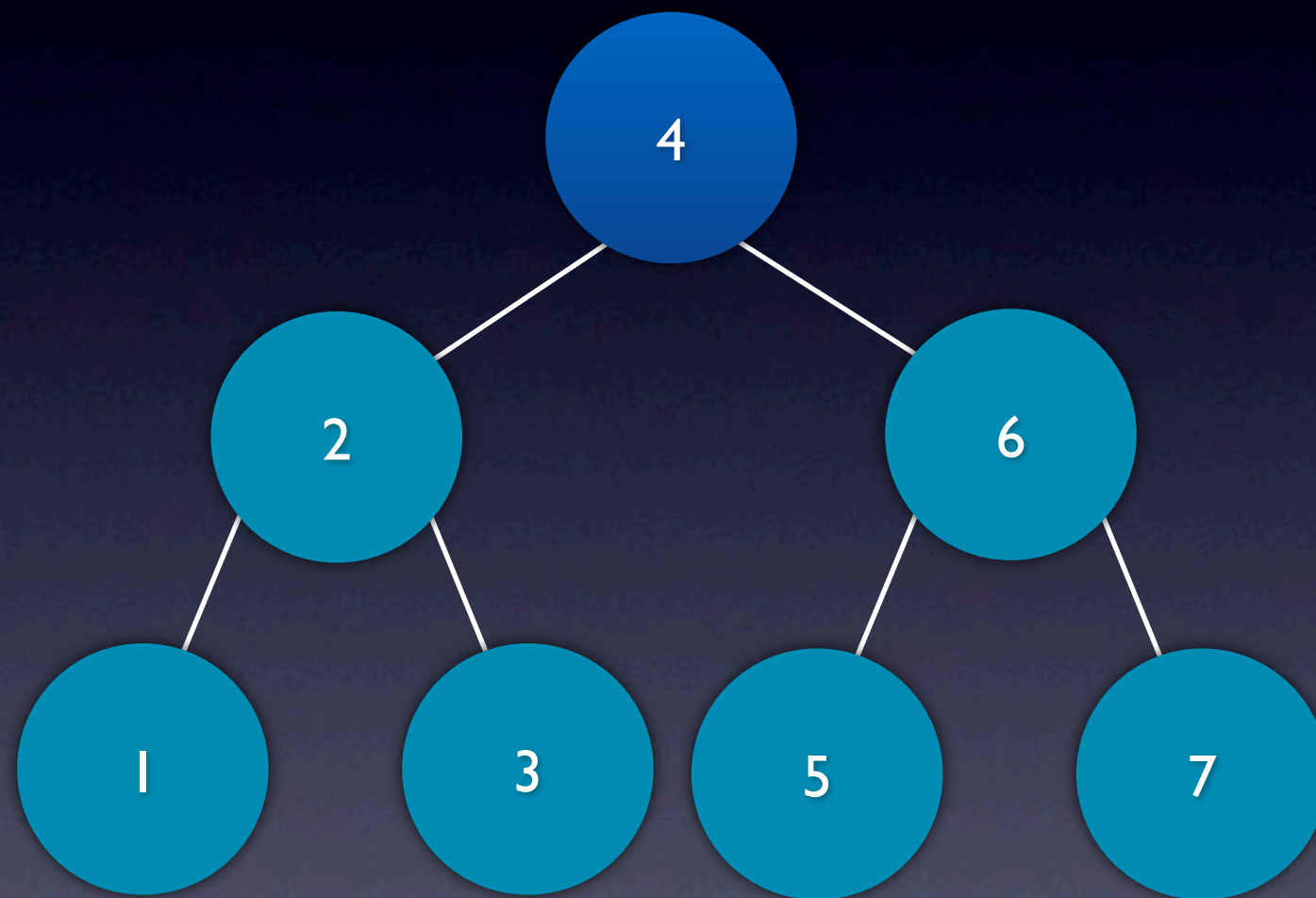
“Besuche den linken Unterbaum, besuche dann die Wurzel, besuche dann den rechten Unterbaum”

Durchlaufen: InOrder



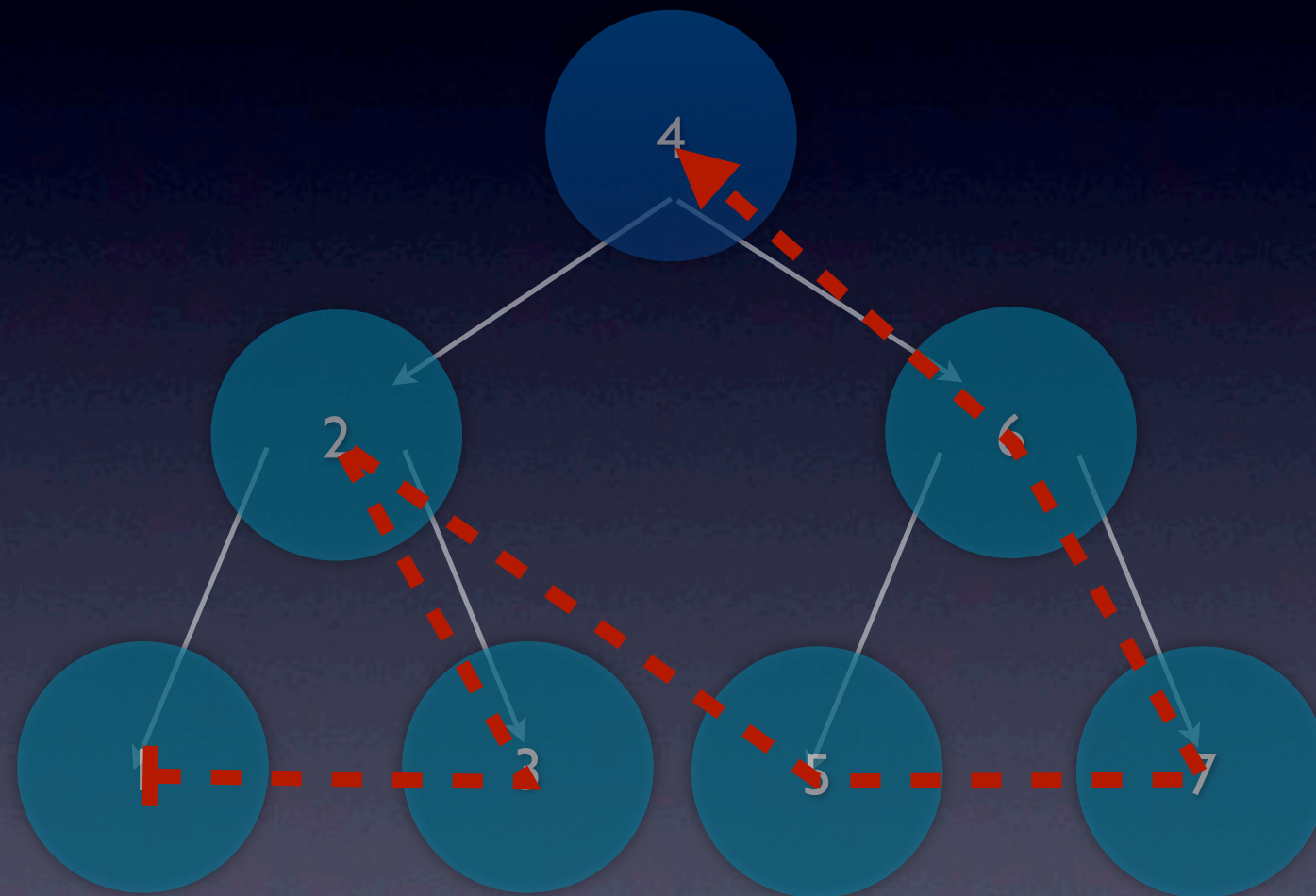
“Besuche den linken Unterbaum, besuche dann die Wurzel, besuche dann den rechten Unterbaum”

Durchlaufen: PostOrder



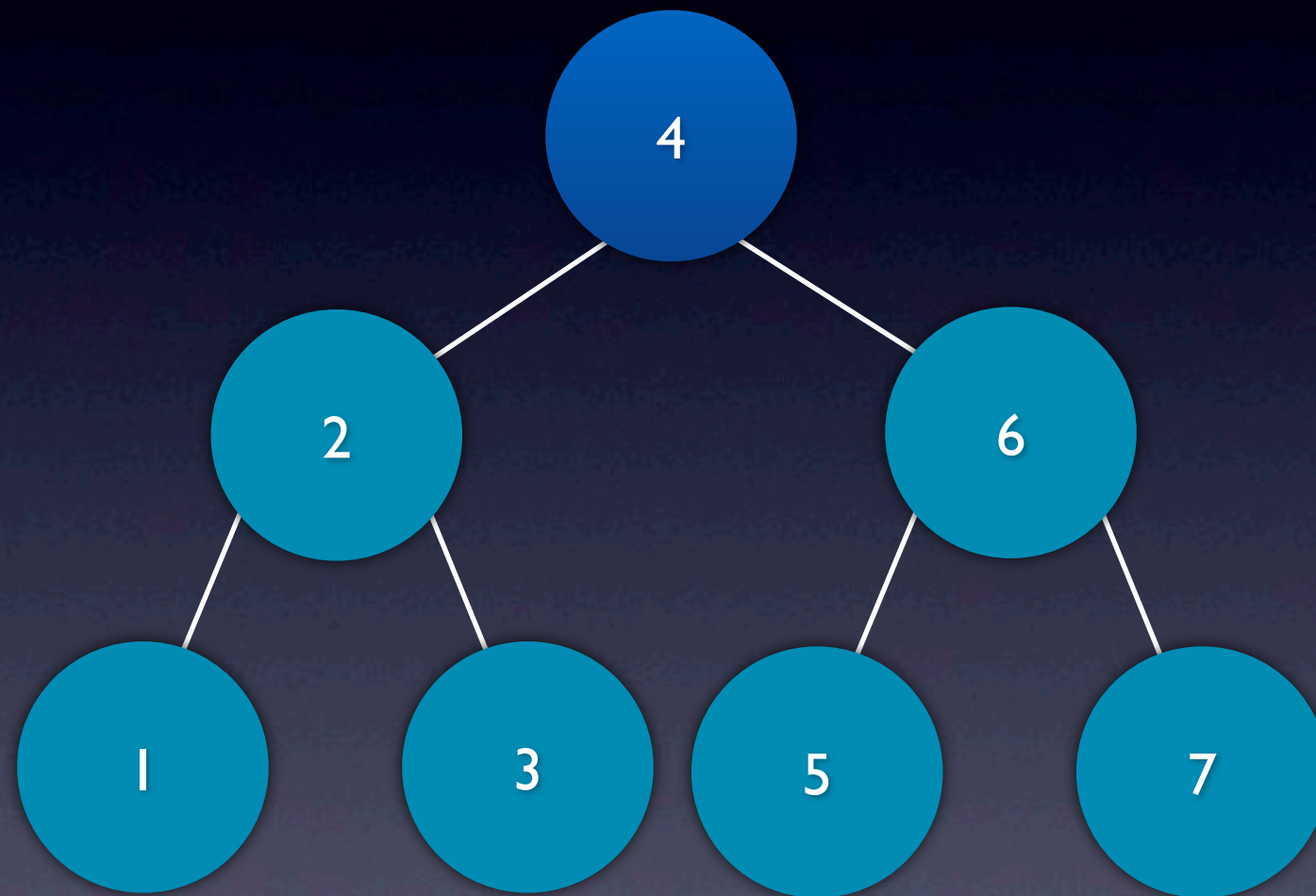
“Besuche den linken Unterbaum, besuche dann den rechten Unterbaum,
besuche dann die Wurzel”

Durchlaufen: PostOrder



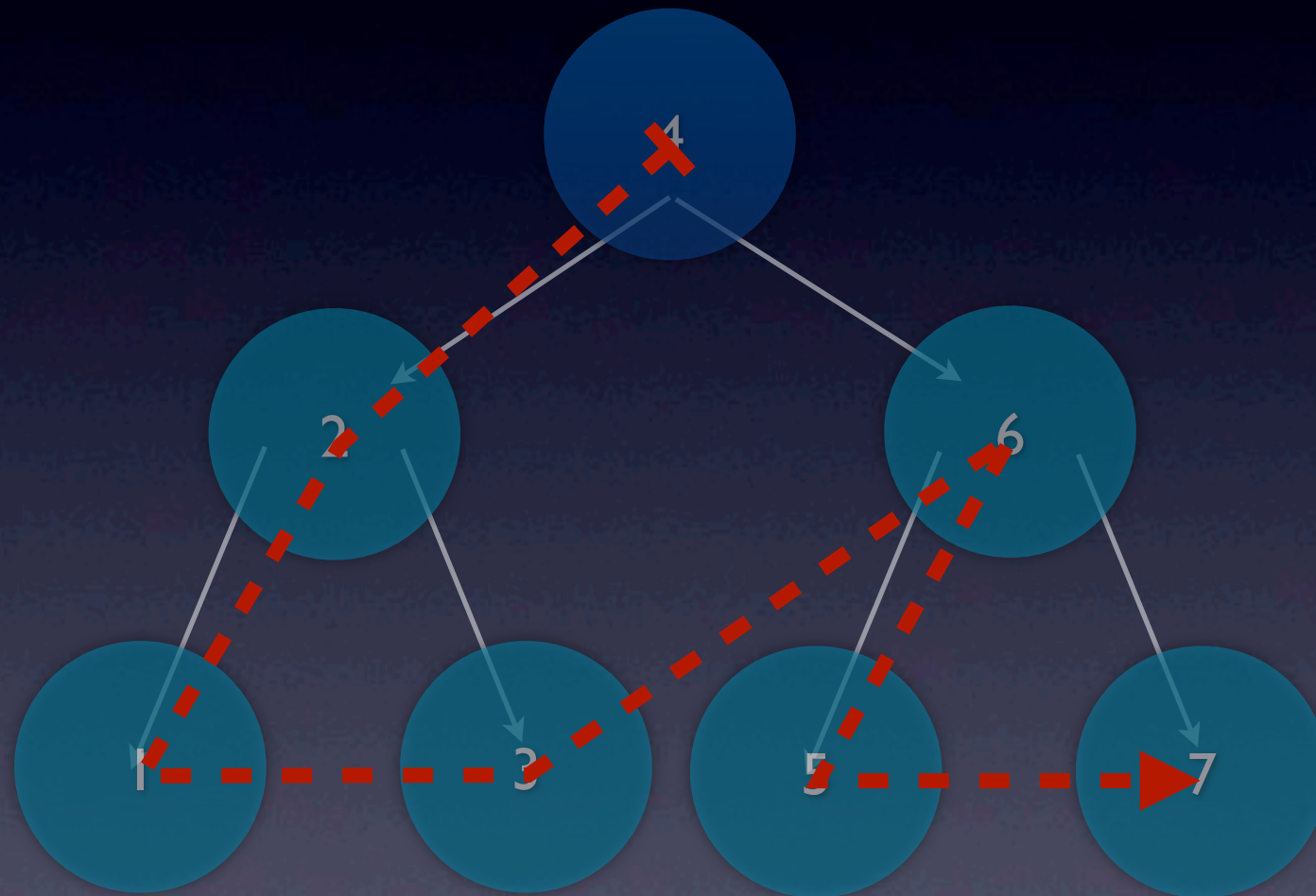
“Besuche den linken Unterbaum, besuche dann den rechten Unterbaum,
besuche dann die Wurzel”

Durchlaufen: PreOrder



“Besuche die Wurzel, besuche dann den linken Unterbaum, besuche dann den rechten Unterbaum”

Durchlaufen: PreOrder



“Besuche die Wurzel, besuche dann den linken Unterbaum, besuche dann den rechten Unterbaum”

Implementation

```
public static void inOrder(TreeNode n) {  
    if(n == null) return;  
    inOrder(n.left);  
    System.out.println(n.value);  
    inOrder(n.right);  
}
```

