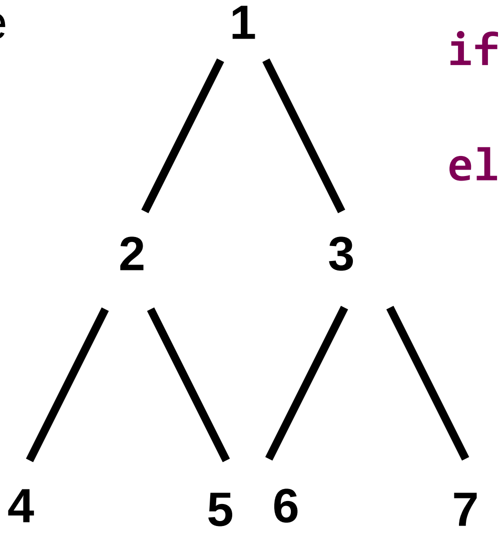
Concurrent Tree Insertion - Problems

When we build a binary tree we want to insert a child on the left edge (if empty) and on the right edge (if left has edge)



```
if(node.left == null)
       node.left = child;
else
       node.right = child;
```

Concurrent Tree Insertion - Problems

Thread 1

```
if(node.left == null)
        node.left = child;
else
        node.right = child;
```

Thread 2

```
if(node.left == null)
        node.left = child;
else
        node.right = child;
```

Concurrent Tree Insertion - Problems

Thread 1

```
if(node.left == null)
        node.left = child;
else
        node.right = child;
```

Thread 2

```
if(node.left == null)
        node.left = child;
else
        node.right = child;
```

Concurrent Tree Insertion - Problems

Thread 1

```
→ if(node.left == null)
          node.left = child;
  else
          node.right = child;
```

Thread 2

```
→ if(node.left == null)
          node.left = child;
  else
          node.right = child;
```

Concurrent Tree Insertion - Problems

Thread 1

```
if(node.left == null)
        node.left = child;
➜ else
        node.right = child;
```

Thread 2

```
if(node.left == null)
➜       node.left = child;
else
        node.right = child;
```

Concurrent Tree Insertion - Problems

Thread 1

```
 if(node.left == null)
        node.left = child;
→else
        node.right = child;
```

Child node for thread 1 is lost

Thread 2

```
 if(node.left == null)
        node.left = child;
→else
        node.right = child;
```

Concurrent Tree Insertion - Problems

- If we use one lock for the entire tree no two threads can insert at the same time
- We can use one lock for each node in the tree

- Keep in mind a tree can insert a node only if the nodes parent is in the tree (Solution: busy waiting)