CSCI 232:
Data Structures and Algorithms

Trees (Part 2)

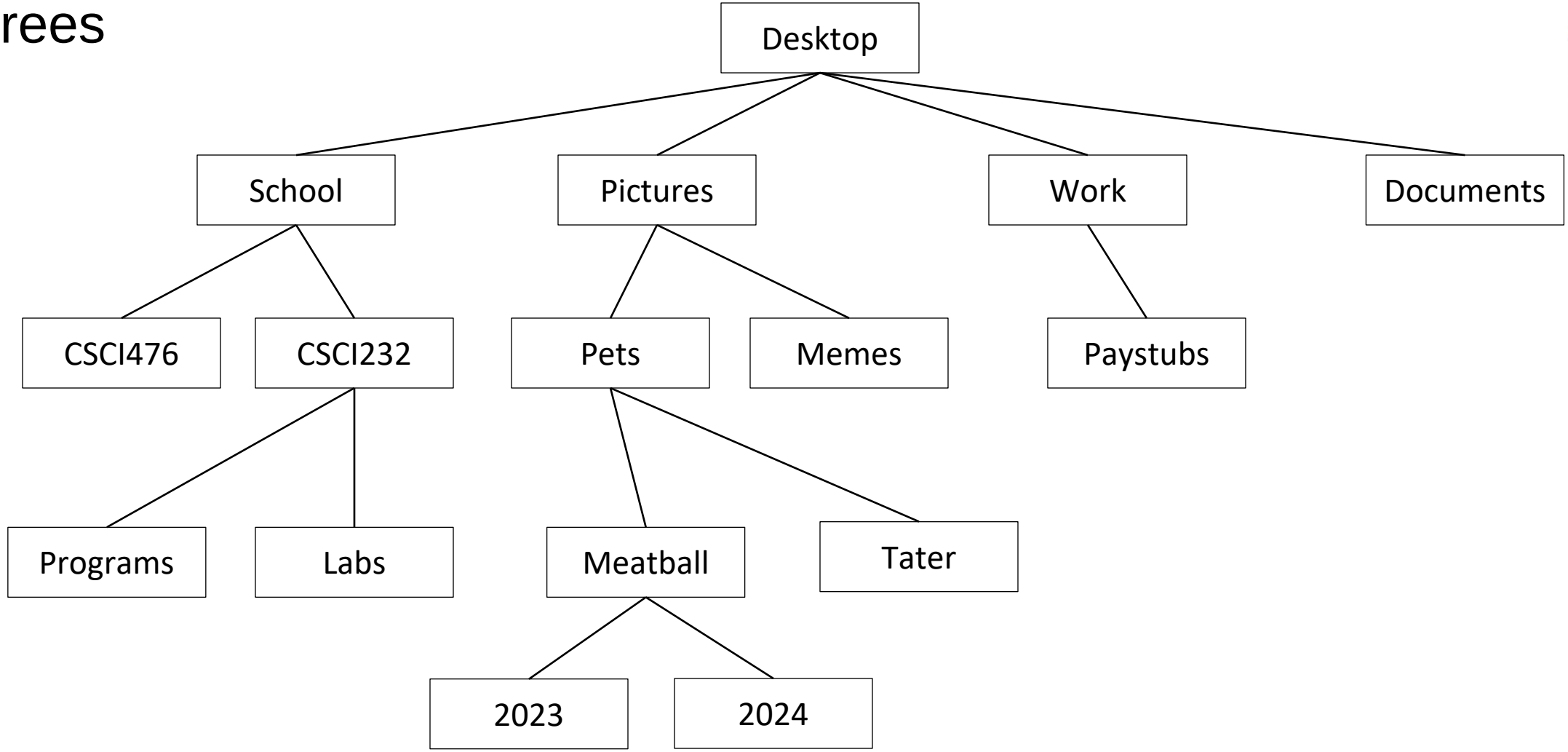
Reese Pearsall

Spring 2024

Announcements

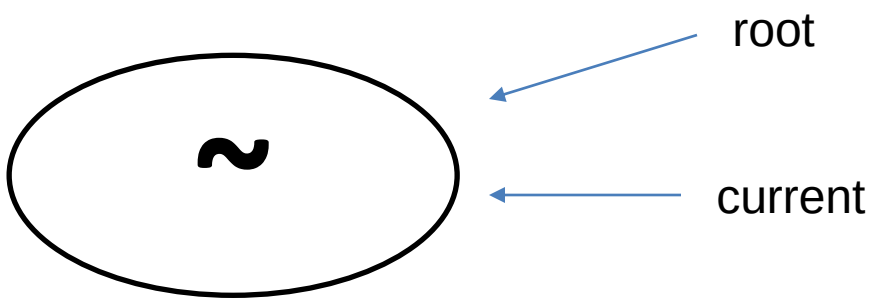
Lab 2 due **tomorrow** at 11:59 PM

Trees

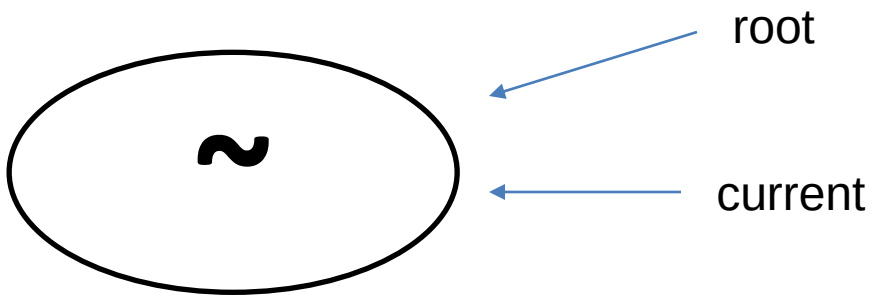


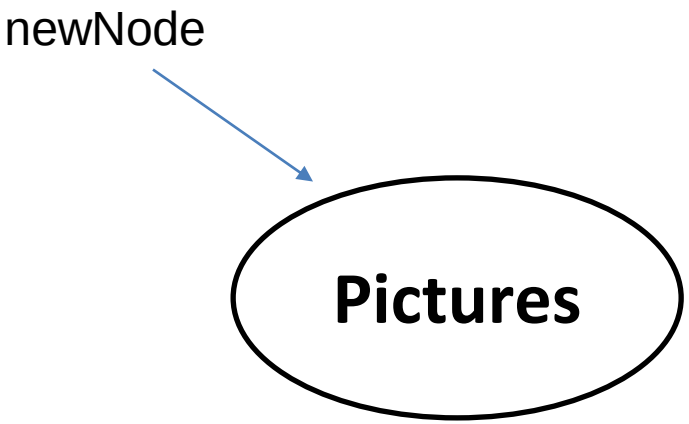
Trees are data structures used to store elements hierarchically
(not linear like arrays and linked lists)

```
public FileTree() {
    this.root = new Node("~");
    this.current = root;
}
```



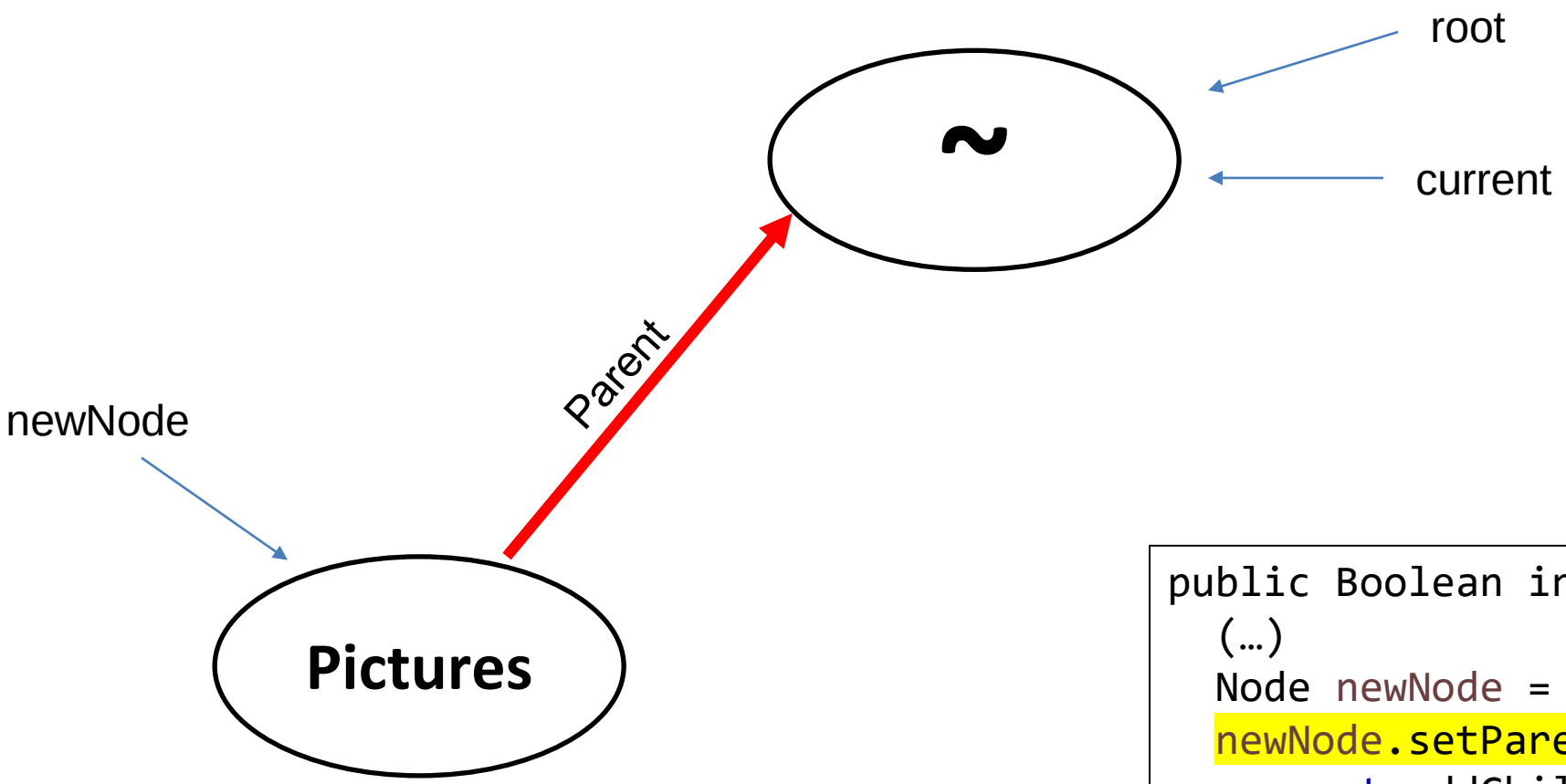
```
public FileTree() {
    this.root = new Node("~");
    this.current = root;
}
```





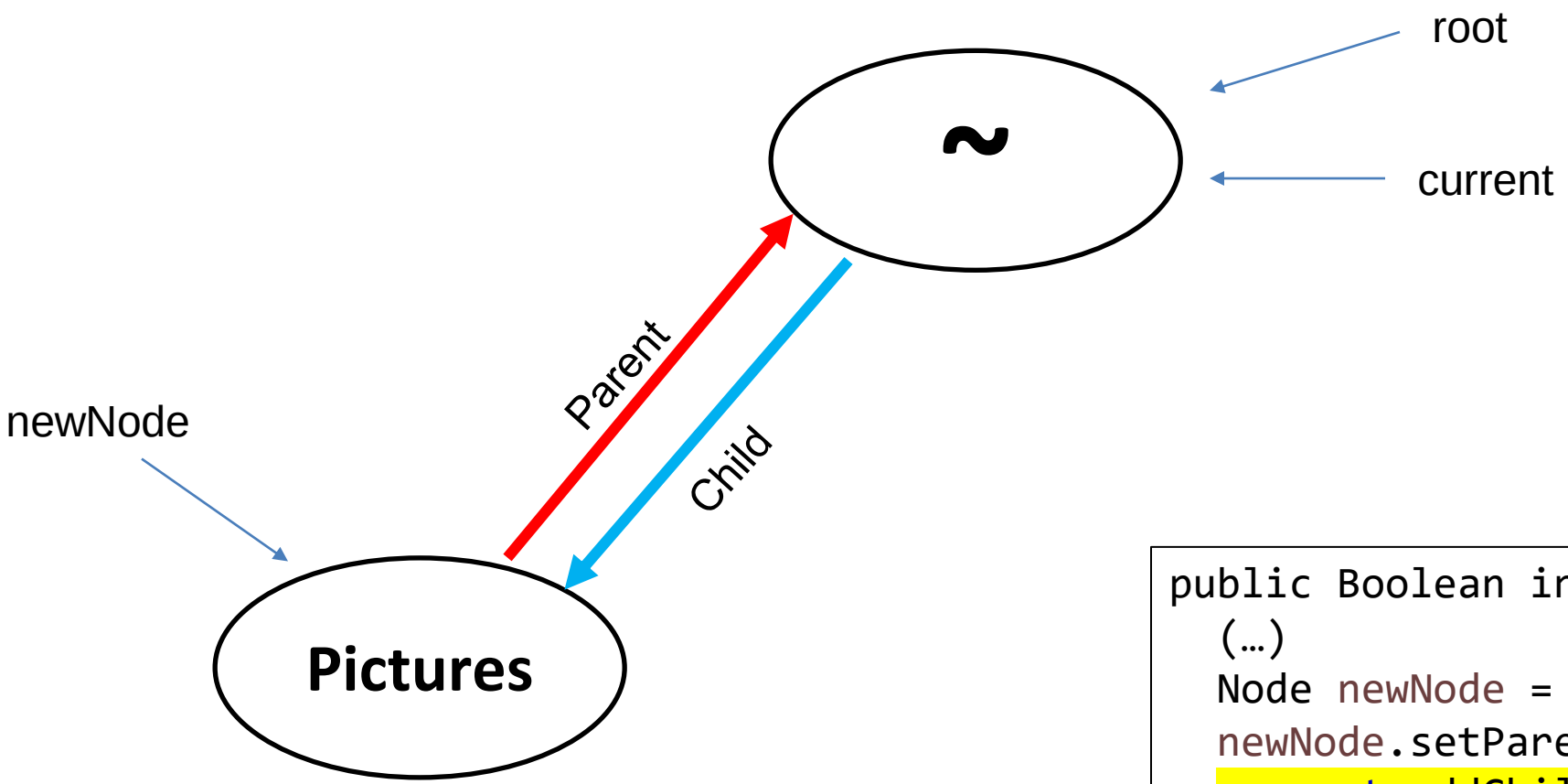
New node that needs to be added?

```
public Boolean insert(String directory){
  (…)
  Node newNode = new Node(directory);
  newNode.setParent(current);
  current.addChild(newNode);
  (…)
}
```



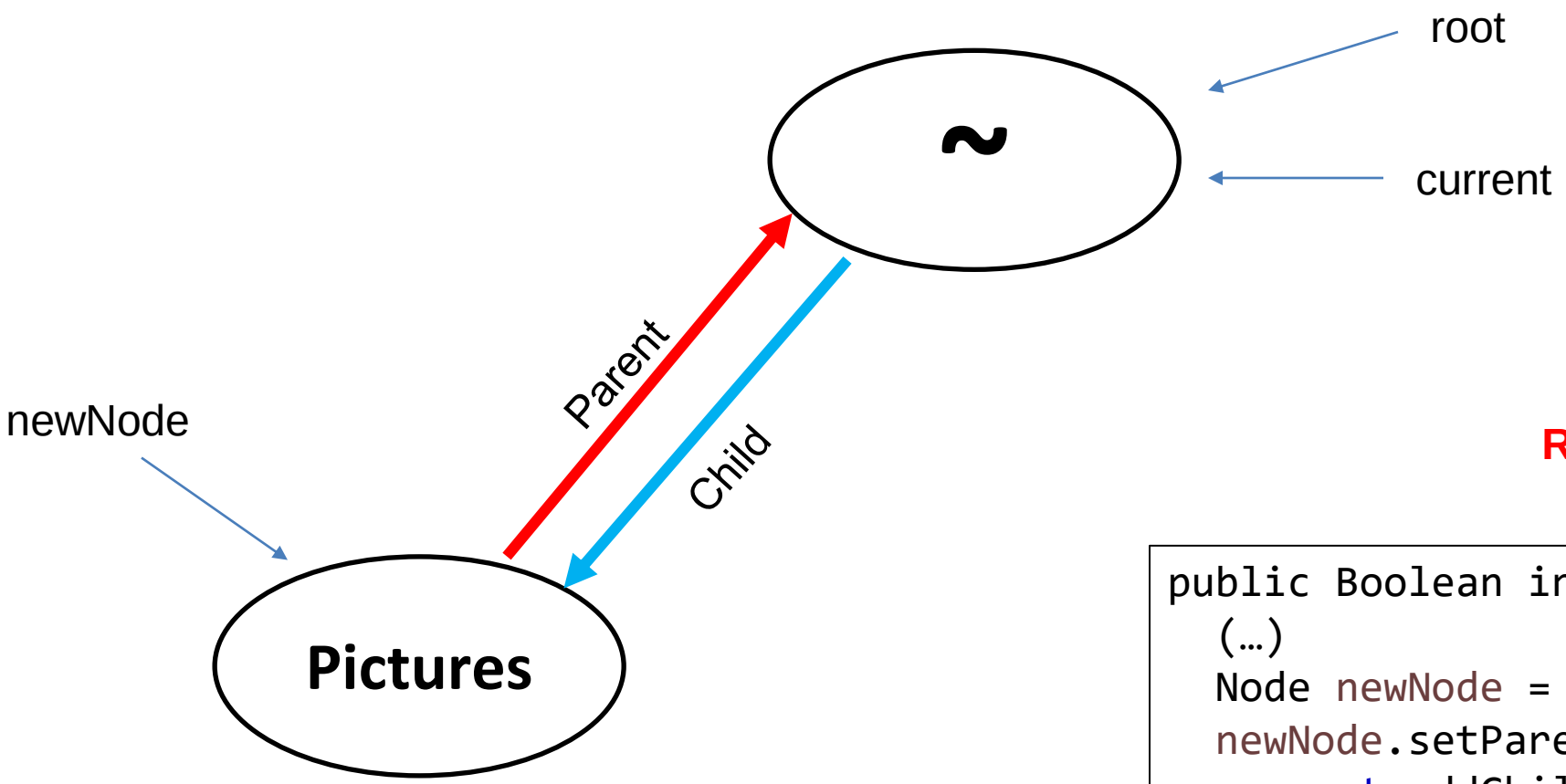
New node that needs to be added?

```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
  (…)
}
```




```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
  (…)
}
```

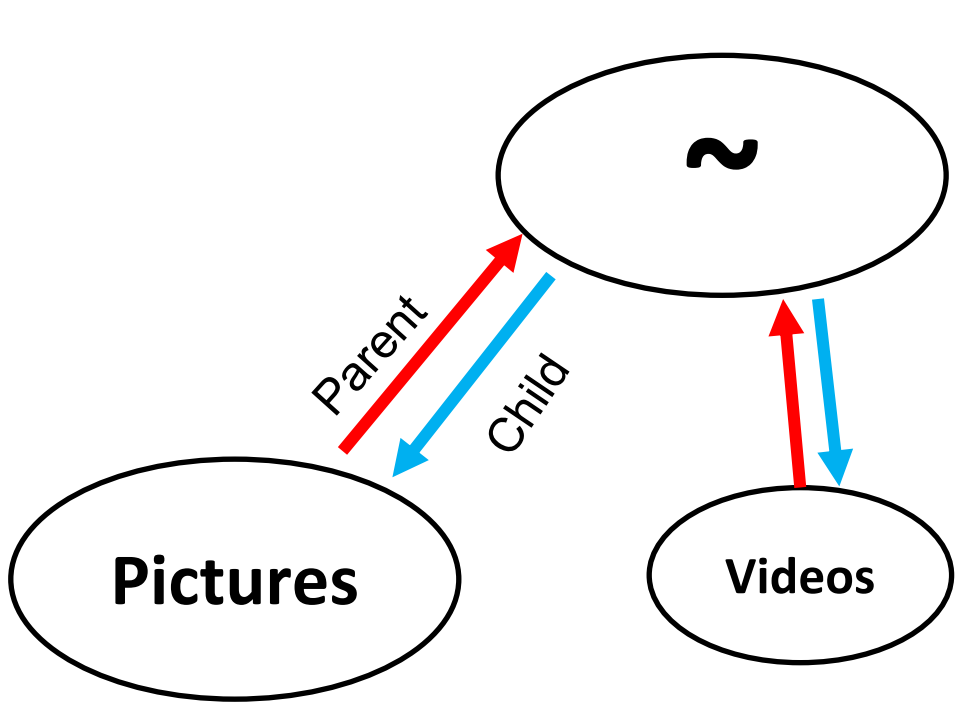
New node that needs to be added?



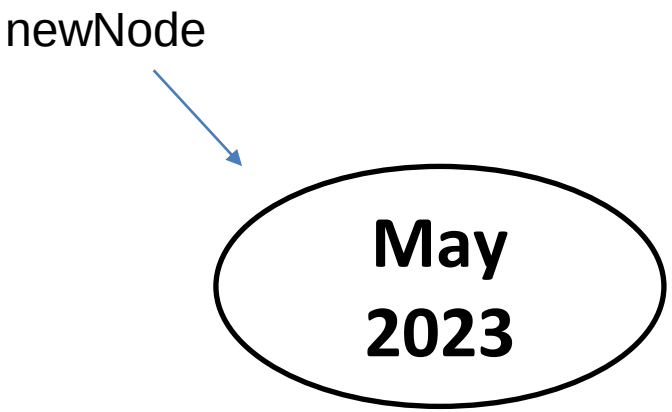
Running time?

```
public Boolean insert(String directory){
  (…)
  Node newNode = new Node(directory);
  newNode.setParent(current);
  current.addChild(newNode);
  (…)
}
```

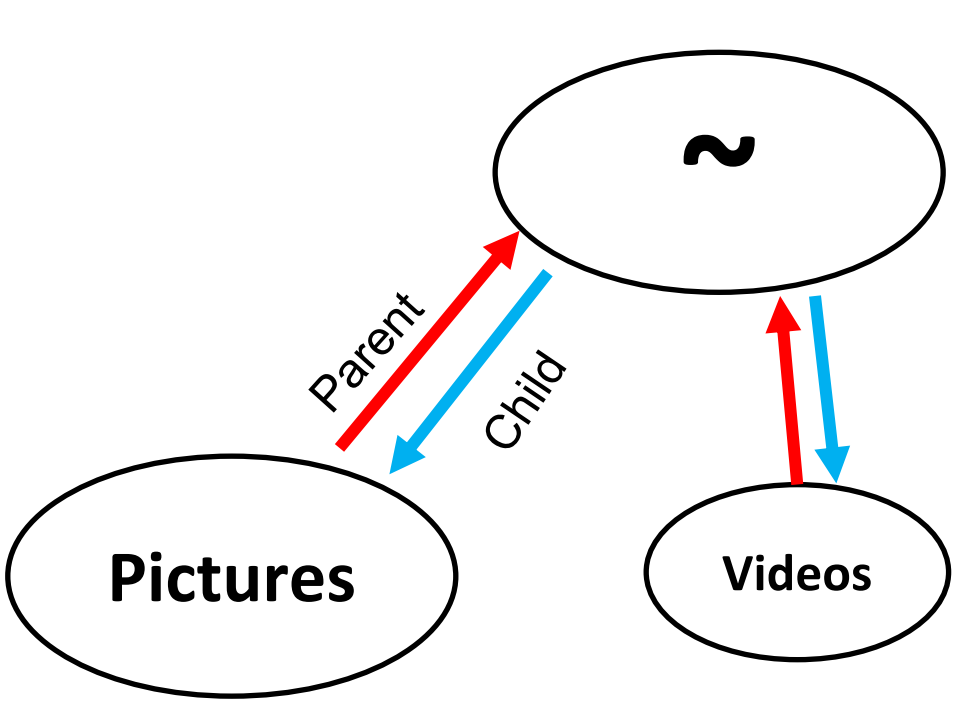
New node that needs to be added?




```
public Boolean insert(String directory){
  (…)
  Node newNode = new Node(directory);
  newNode.setParent(current);
  current.addChild(newNode);
 (…)
}
```

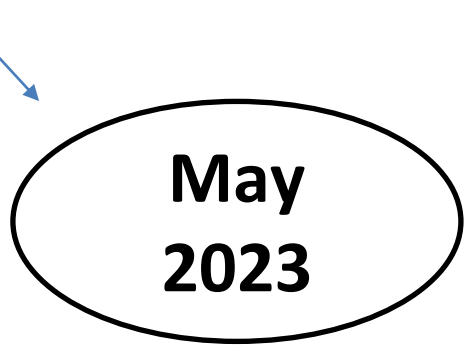


```
tree.moveDown("Pictures");
```



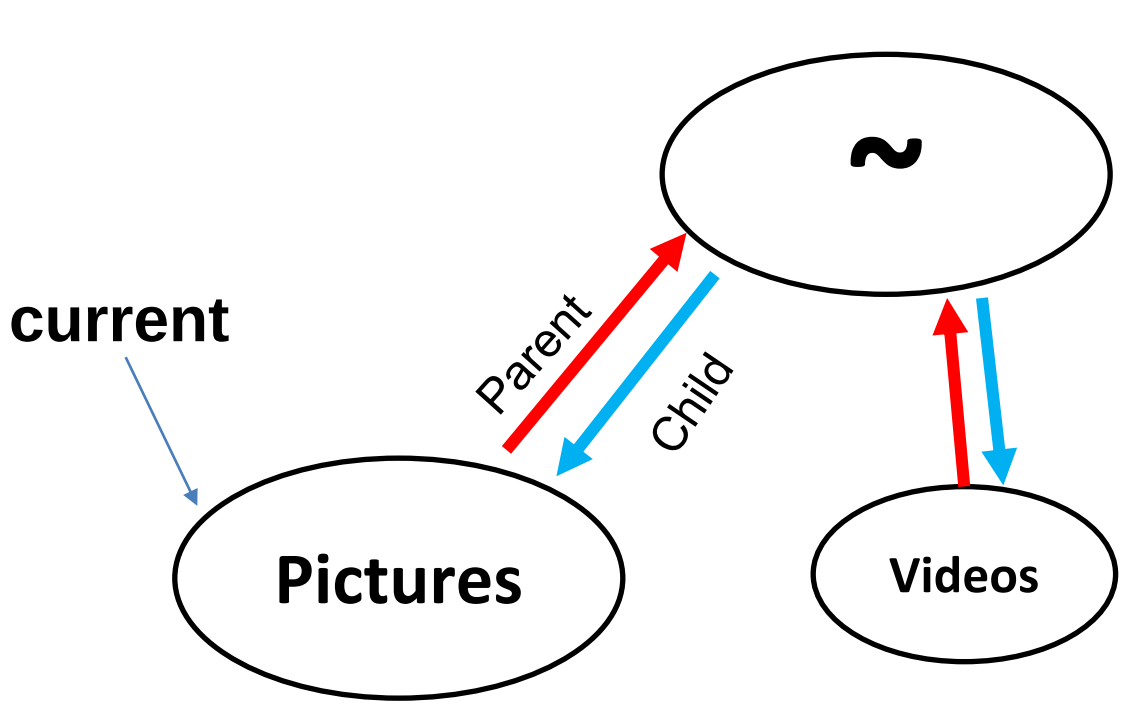

```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
  (…)
}
```

newNode

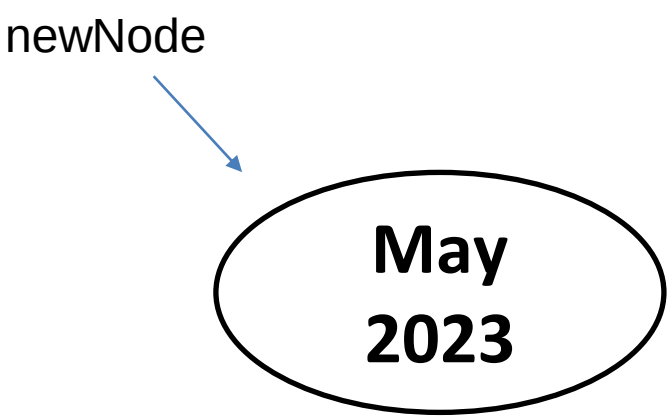



```
public boolean moveDown(String directory) {
   ArrayList<Node> children = current.getChildren();
      for(Node child: children) {
         if(child.getName().equals(directory)) {
            current = child;
            return true;
         }
   }
    return false;
}
```

```
tree.moveDown("Pictures");
```

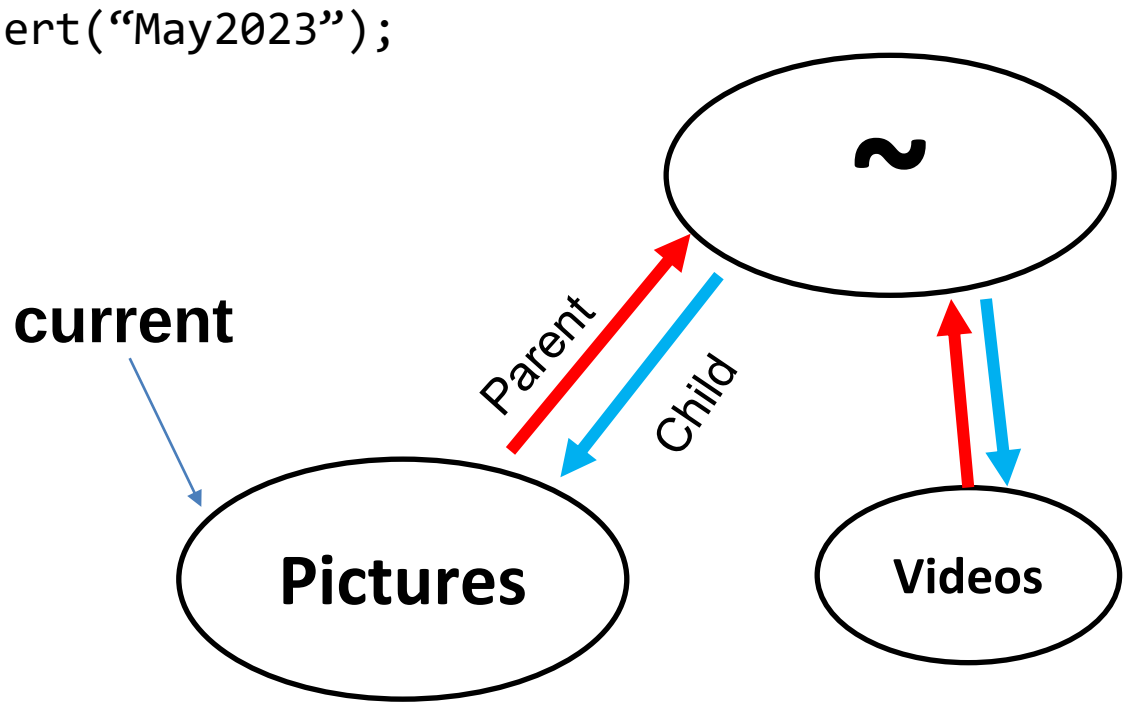



```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
  (…)
}
```




```
public boolean moveDown(String directory) {
   ArrayList<Node> children = current.getChildren();
      for(Node child: children) {
         if(child.getName().equals(directory)) {
            current = child;
            return true;
         }
   }
    return false;
}
```

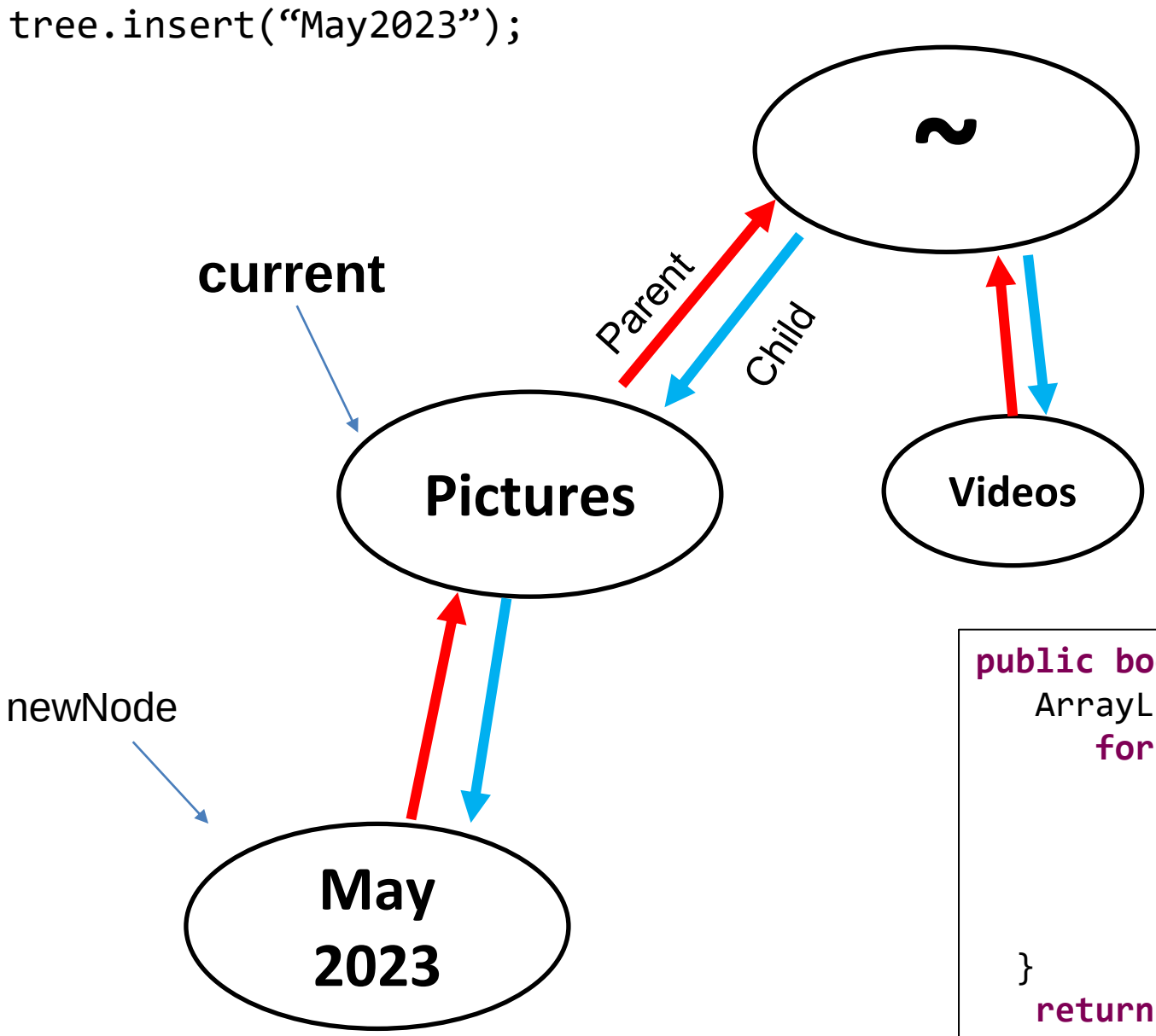
```
tree.moveDown("Pictures");
tree.insert("May2023");
```




```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
  (…)
}
```

```
public boolean moveDown(String directory) {
   ArrayList<Node> children = current.getChildren();
      for(Node child: children) {
         if(child.getName().equals(directory)) {
            current = child;
            return true;
         }
   }
    return false;
}
```

```
tree.moveDown("Pictures");
tree.insert("May2023");
```

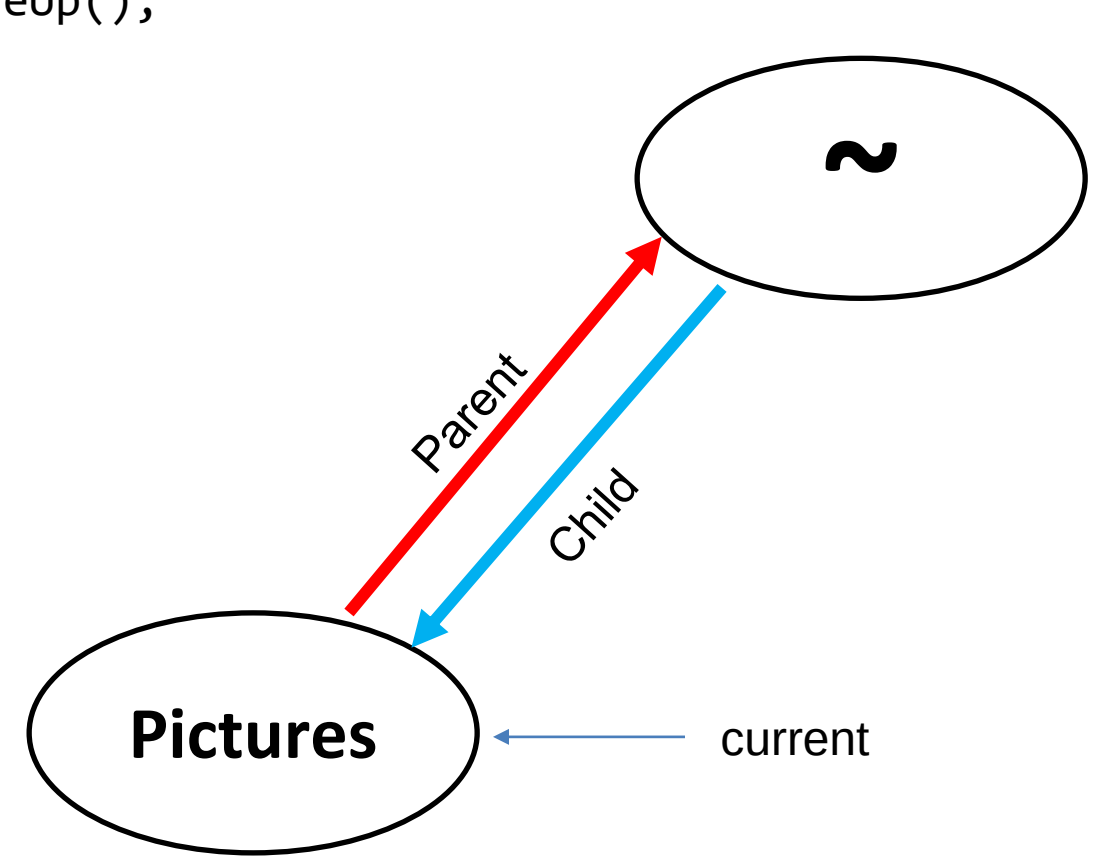



```
public Boolean insert(String directory){
   (…)
   Node newNode = new Node(directory);
   newNode.setParent(current);
   current.addChild(newNode);
   (…)
}
```

```
public boolean moveDown(String directory) {
   ArrayList<Node> children = current.getChildren();
      for(Node child: children) {
         if(child.getName().equals(directory)) {
            current = child;
            return true;
         }
   }
   return false;
}
```

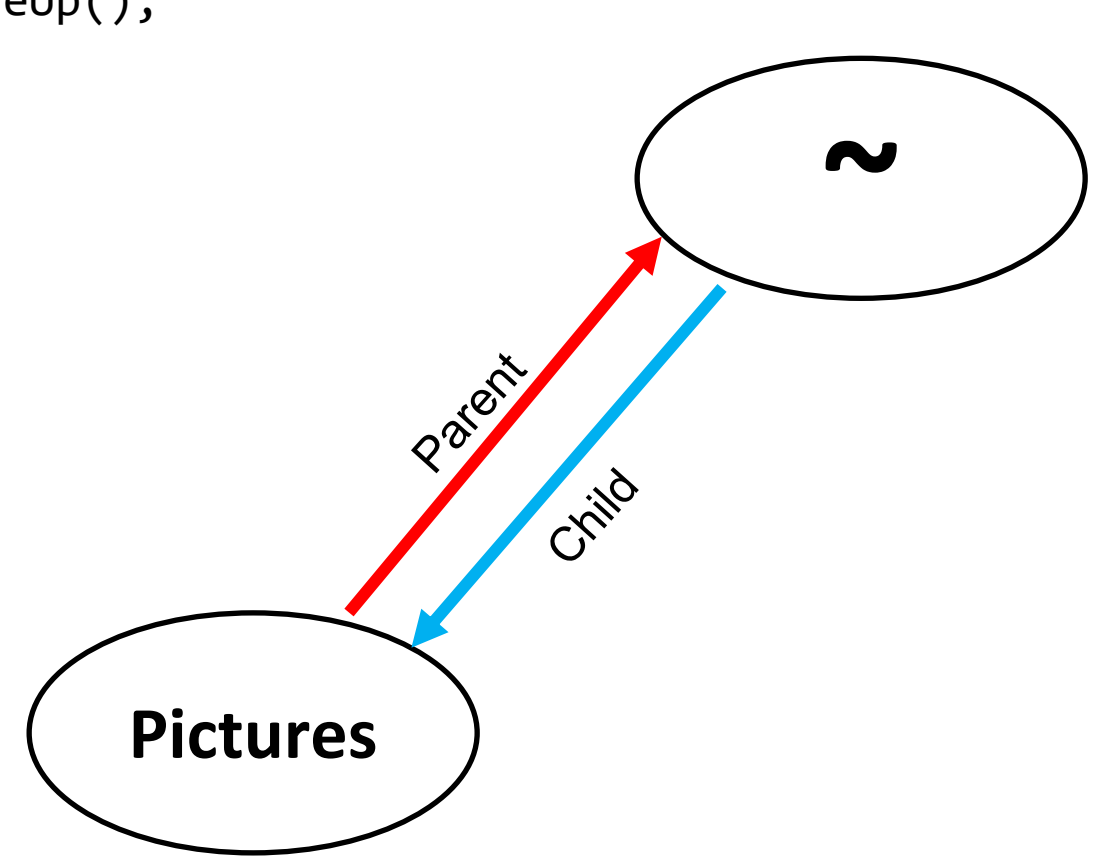
Running time?

```
tree.moveUp();
```



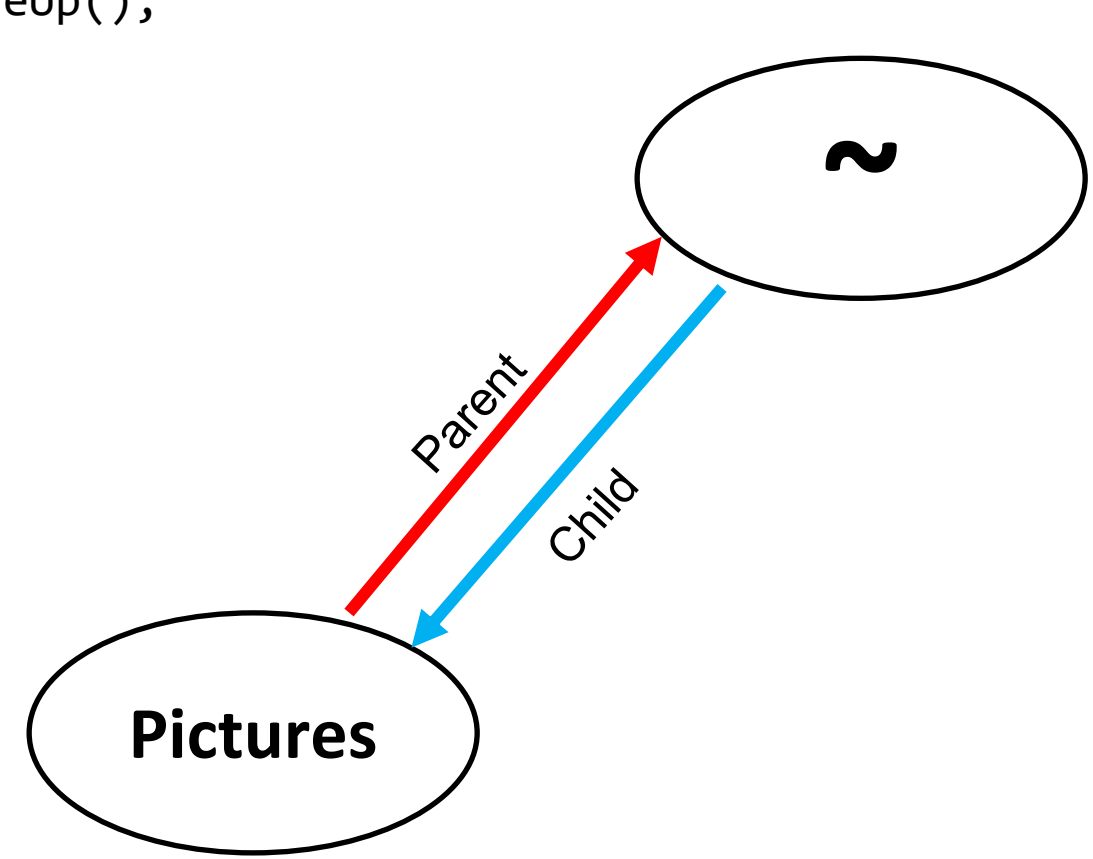

```
public void moveUp() {
    if( current != root ) {
        current = current.getParent();
    }
}
```

```
tree.moveUp();
```




```
public void moveUp() {
    if( current != root ) {
        current = current.getParent();
    }
}
```

```
tree.moveUp();
```




```
public void moveUp() {
    if( current != root ) {
        current = current.getParent();
    }
}
```

Running time?

Running time?

```
public void goHome() {
    current = root;
}
```