Left Rotation: Source Code

Josiah Yoder

For interview at

Milwaukee School of Engineering

8 Apr 2013

Rotate Left

```
public void rotateLeft(Node oldParent) {
    if(null == oldParent) {
        return;
    }
    Node newParent = oldParent._right;
        if(null == newParent) {
            return;
    }
    if(_root == oldParent) {
        _root = newParent;
    }
    newParent._parent = oldParent._parent;
    if(null != newParent._parent) {
        if(newParent._parent._right == oldParent) {
            newParent._parent._right = newParent;
        }
        else {
            newParent._parent._left = newParent;
        }
    }
    oldParent._right = newParent._left;
    if(null != oldParent._right) {
        oldParent._right._parent = oldParent;
    }
    newParent._left = oldParent;
    oldParent._parent = newParent;
}
```

Input Checking

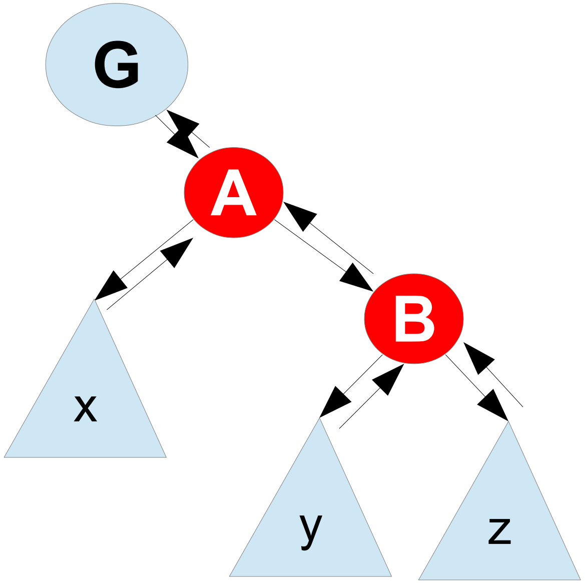
Link B to A's parent

Link B's left child to A

Link A to B

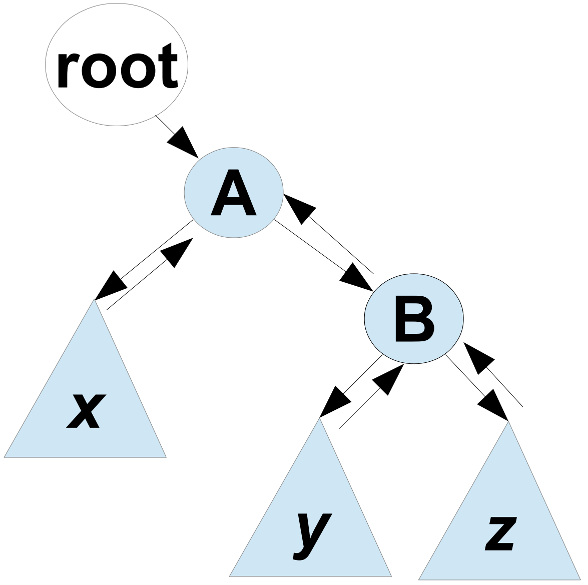
Input checking

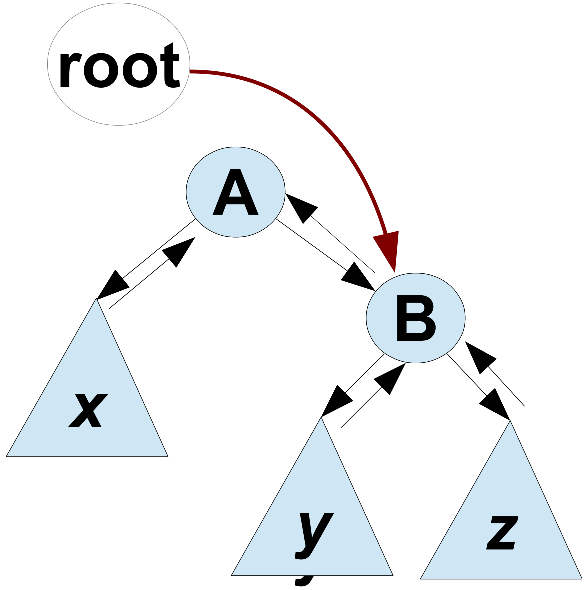
```
public void rotateLeft(Node oldParent) {
    if(null == oldParent) {
        return;
    }
    Node newParent = oldParent._right;
    if(null == newParent) {
            return;
    }
...
```



Link B to A's Parent (1)

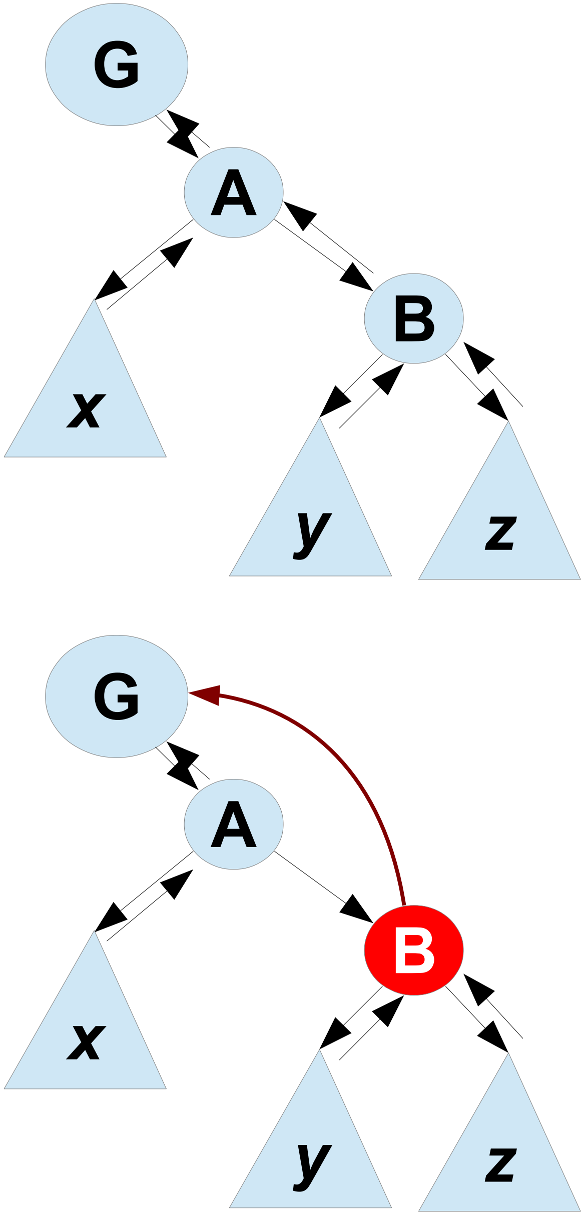
```
if(_root == oldParent) {
    _root = newParent;
}
newParent._parent = oldParent._parent;
if(null != newParent._parent) {
    if(newParent._parent._right == oldParent) {
        newParent._parent._right = newParent;
    }
    else {
        newParent._parent._left = newParent;
    }
}
```





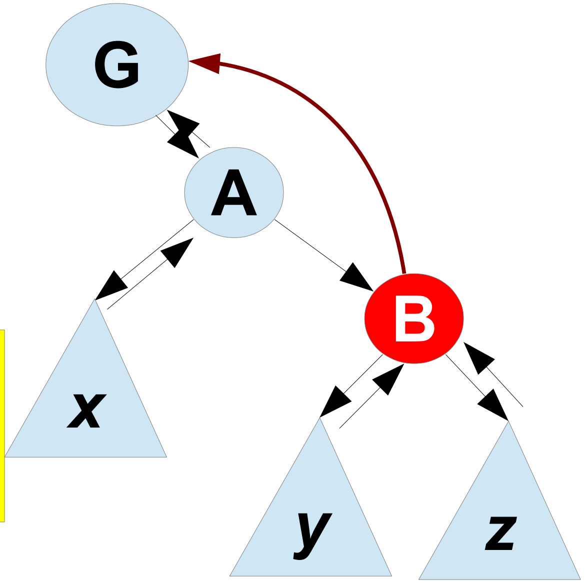
Link B to A's Parent (2)

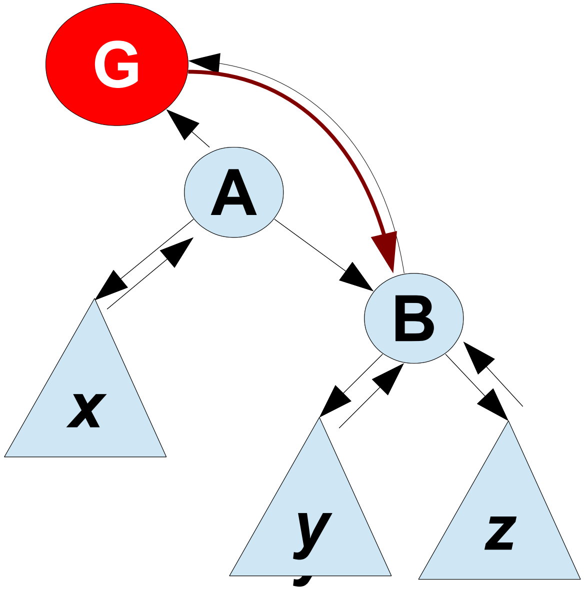
```
if(_root == oldParent) {
    _root = newParent;
}
newParent._parent = oldParent._parent;
if(null != newParent._parent) {
    if(newParent._parent._right == oldParent) {
        newParent._parent._right = newParent;
    }
    else {
        newParent._parent._left = newParent;
    }
}
```



Link B to A's Parent (3)

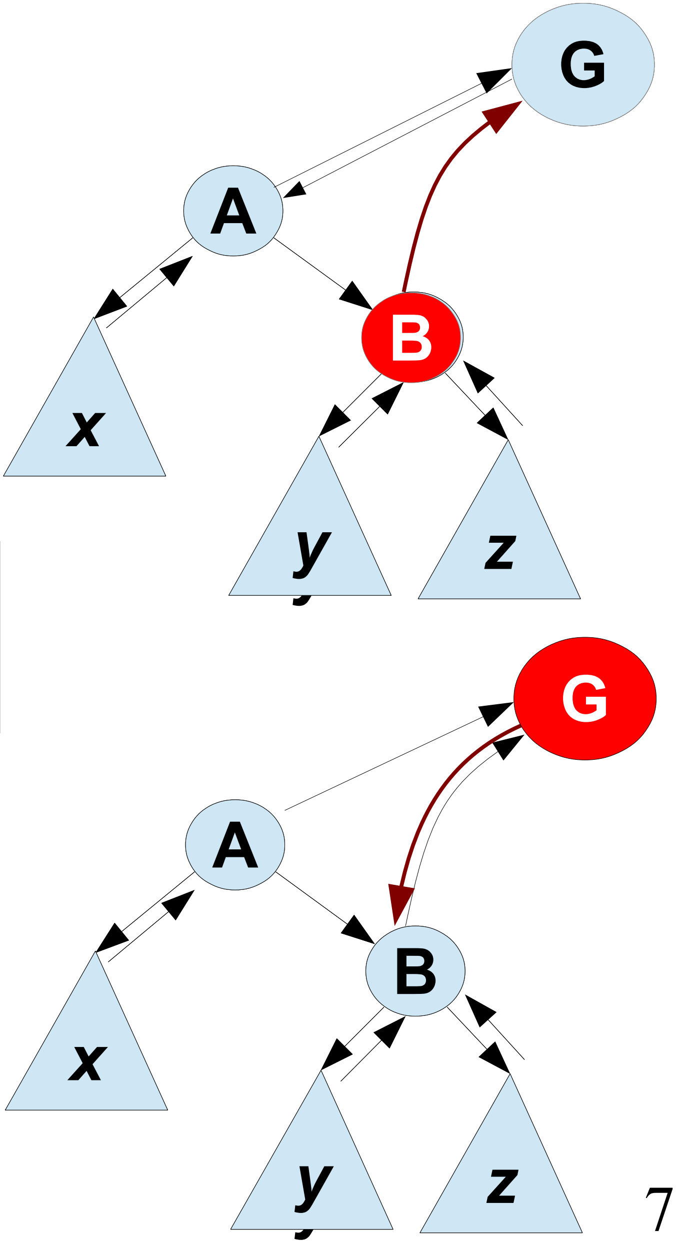
```
if(_root == oldParent) {
    _root = newParent;
}
newParent._parent = oldParent._parent;
if(null != newParent._parent) {
    if(newParent._parent._right == oldParent) {
        newParent._parent._right = newParent;
    }
    else {
        newParent._parent._left = newParent;
    }
}
```





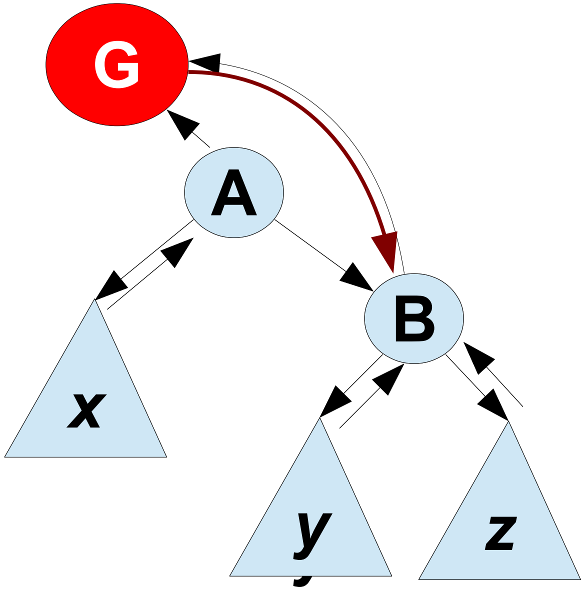
Link B to A's Parent (4)

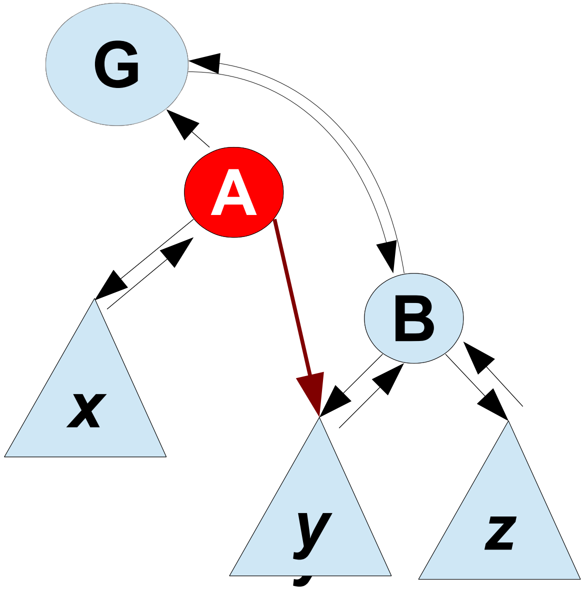
```
if(_root == oldParent) {
    _root = newParent;
}
newParent._parent = oldParent._parent;
if(null != newParent._parent) {
    if(newParent._parent._right == oldParent) {
        newParent._parent._right = newParent;
    }
    else {
        newParent._parent._left = newParent;
    }
}
```



Link B's Left Child to A (1)

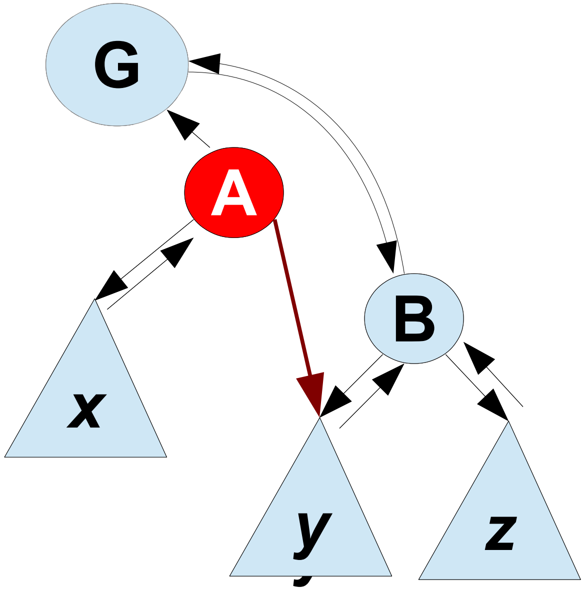
```
oldParent._right = newParent._left;
if(null != oldParent._right) {
    oldParent._right._parent = oldParent;
}
```

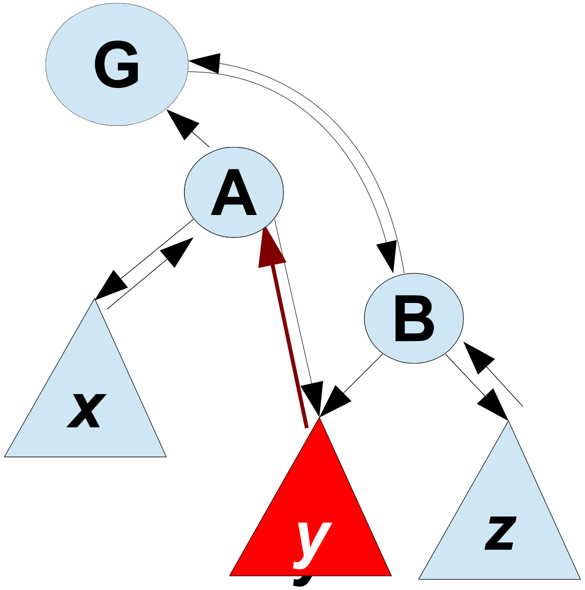




Link B's Left Child to A (2)

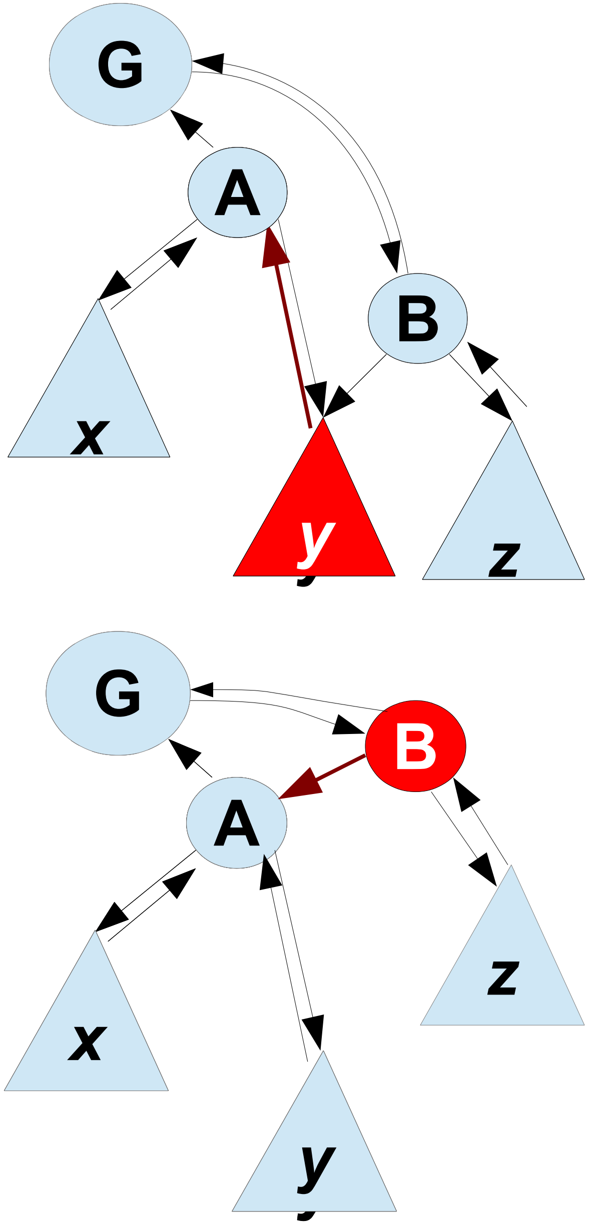
```
oldParent._right = newParent._left;
if(null != oldParent._right) {
    oldParent._right._parent = oldParent;
}
```





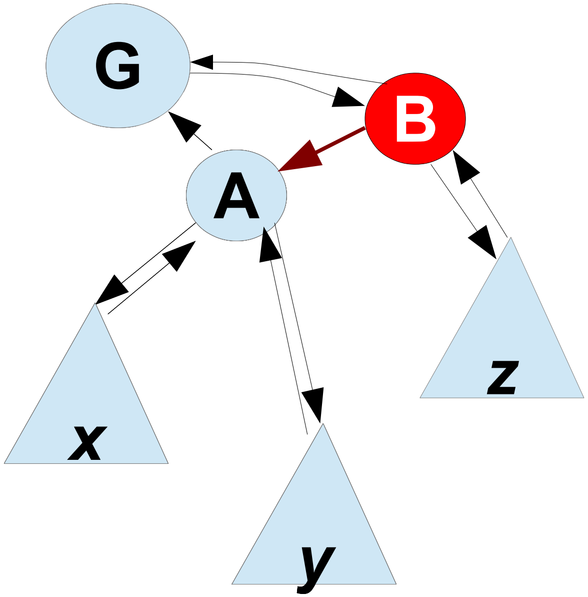
Link A to B (1)

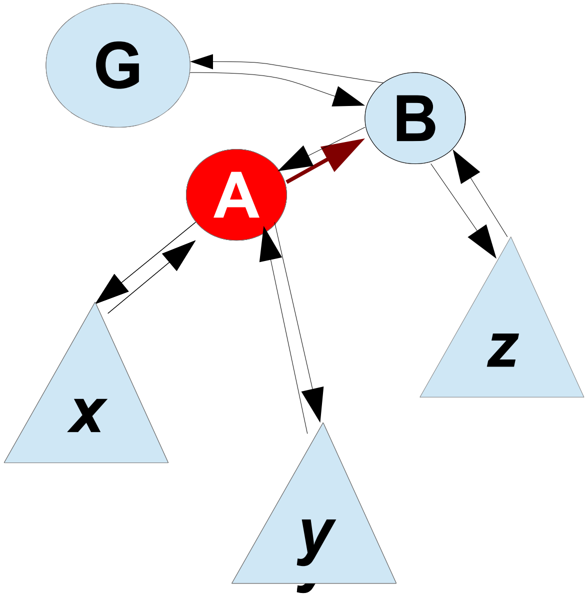
```
...
    newParent._left = oldParent;
    oldParent._parent = newParent;
}
```



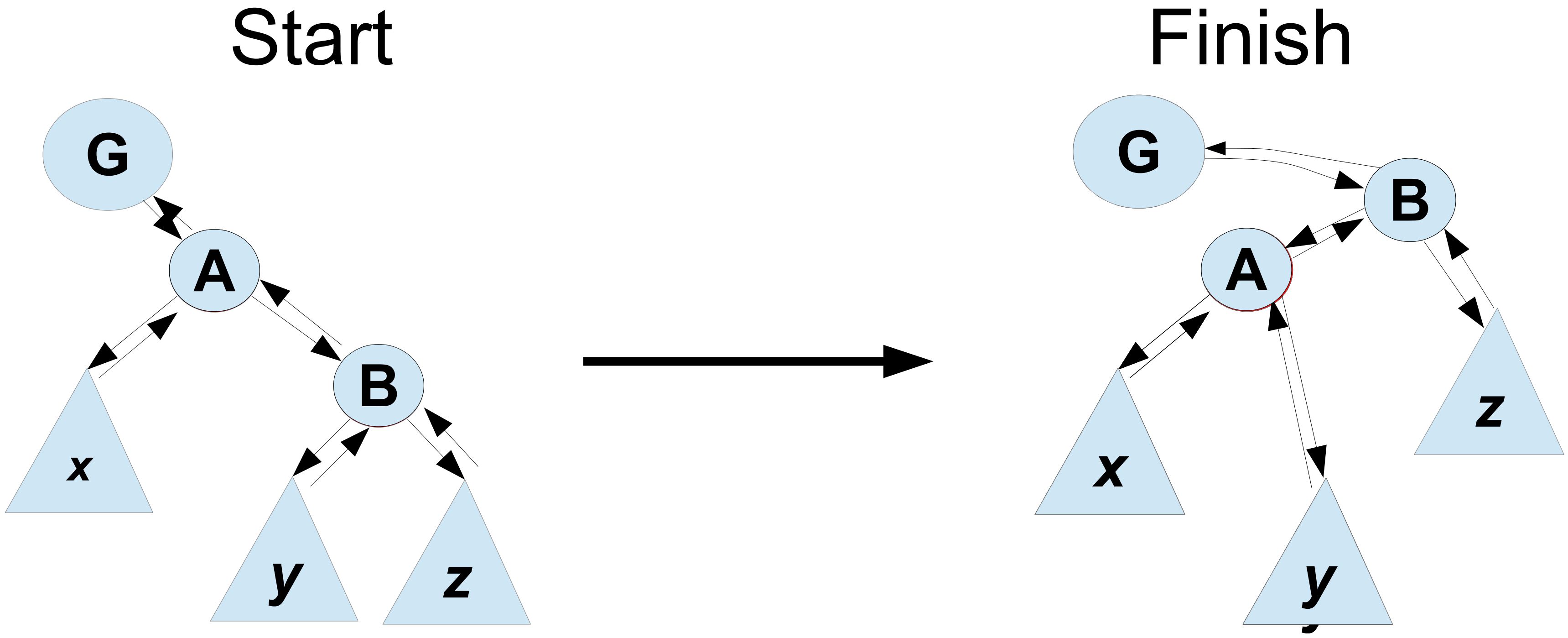
Link A to B (2)

```
...
    newParent._left = oldParent;
    oldParent._parent = newParent;
}
```





Complete Left Rotation


Start
G
A
B
x
y
z
Finish
G
B
A
x
y
z