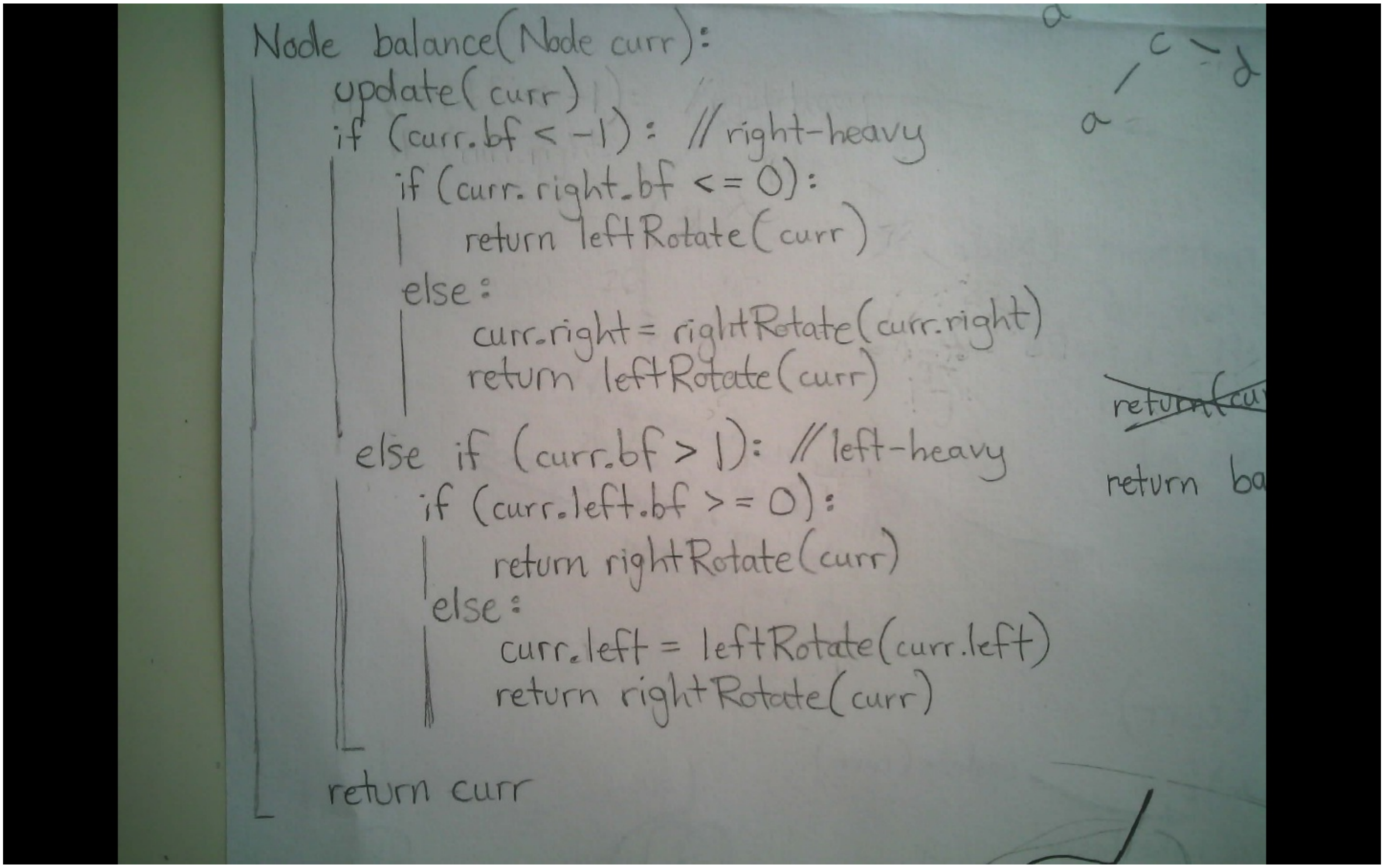
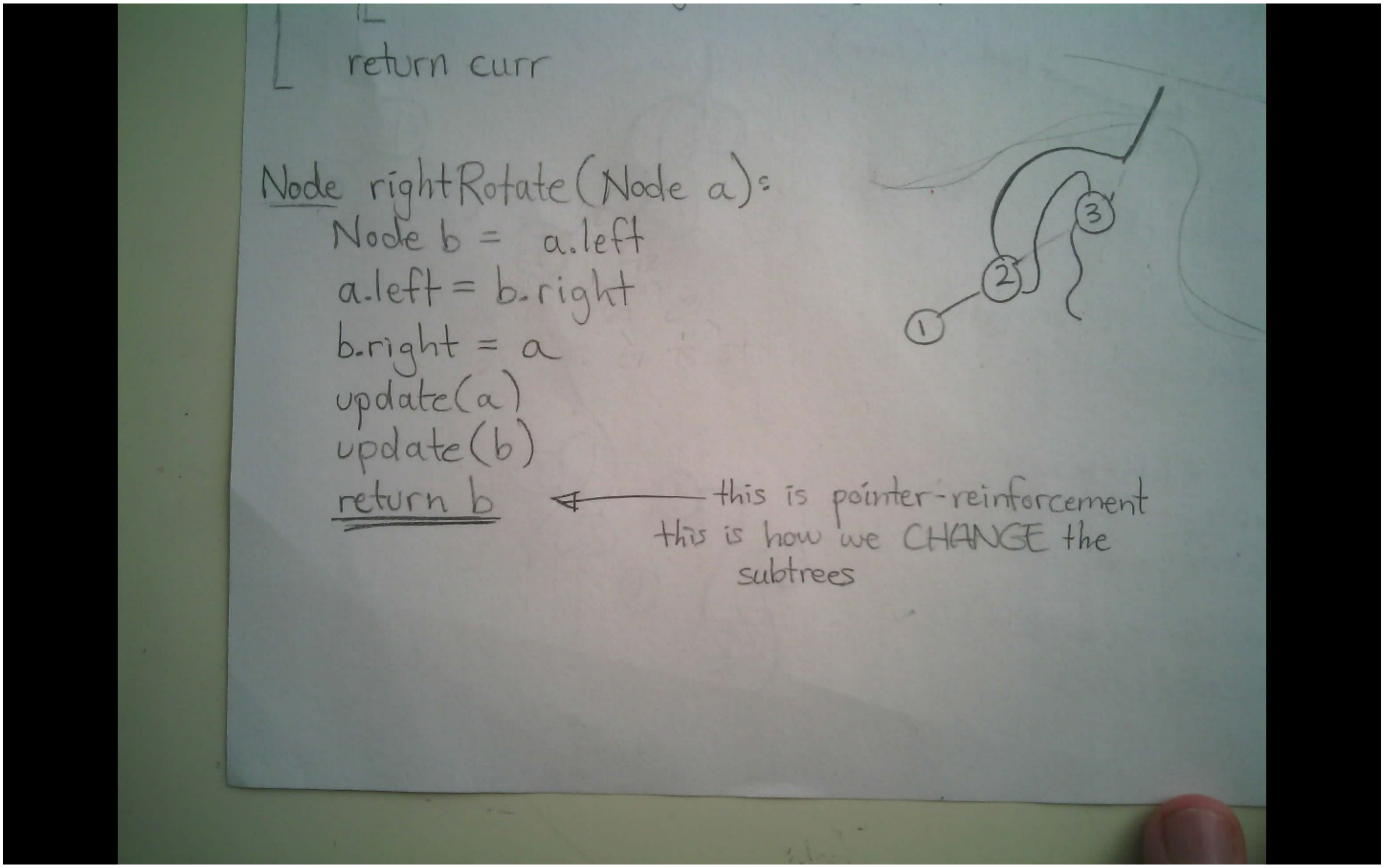


```
Node balance(Node curr):
    update(curr)
    if (curr.bf < -1): //right-heavy
        if (curr.right.bf <= 0):
            return leftRotate(curr)
        else:
            curr.right = rightRotate(curr.right)
            return leftRotate(curr)
    else if (curr.bf > 1): //left-heavy
        if (curr.left.bf >= 0):
            return rightRotate(curr)
        else:
            curr.left = leftRotate(curr.left)
            return rightRotate(curr)
    return curr
```

~~return(cu~~

return ba



```
    return curr

Node rightRotate(Node a):
    Node b =  a.left
    a.left = b.right
    b.right = a
    update(a)
    update(b)
    return b
```

this is pointer-reinforcement
this is how we CHANGE the subtrees