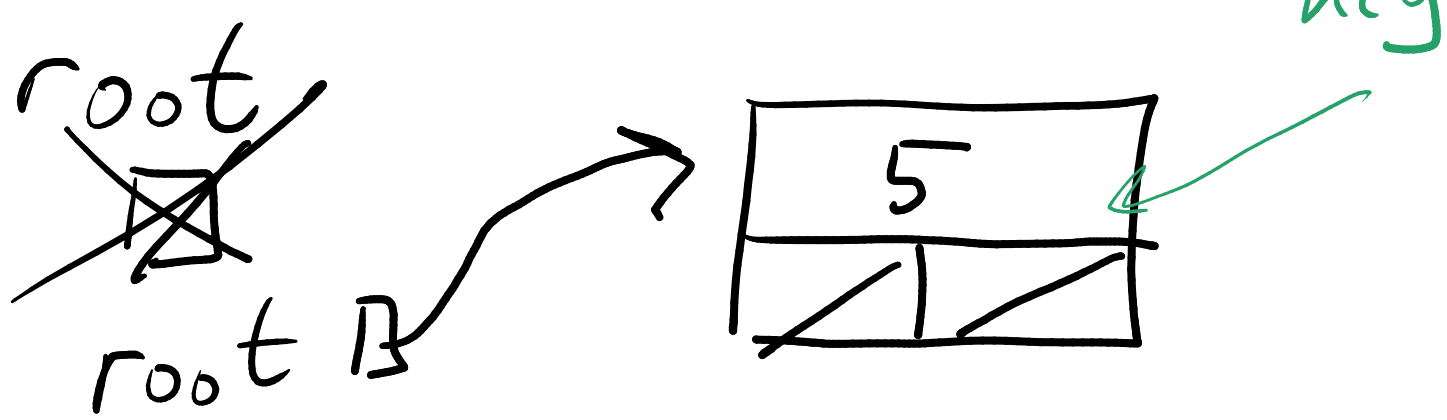
# Binary search trees - insertion, deletion

## insertion

5

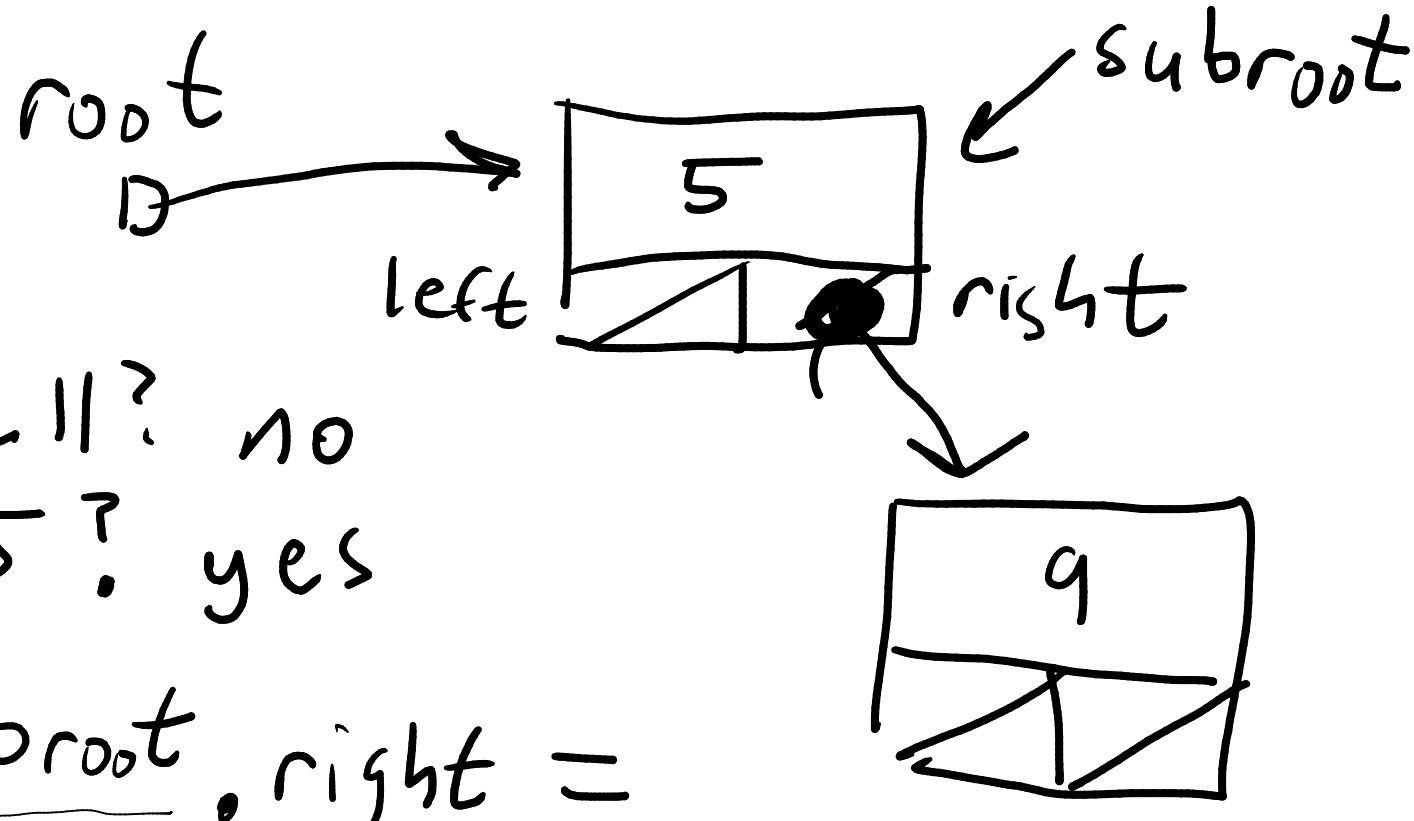
insert




---

9

insert

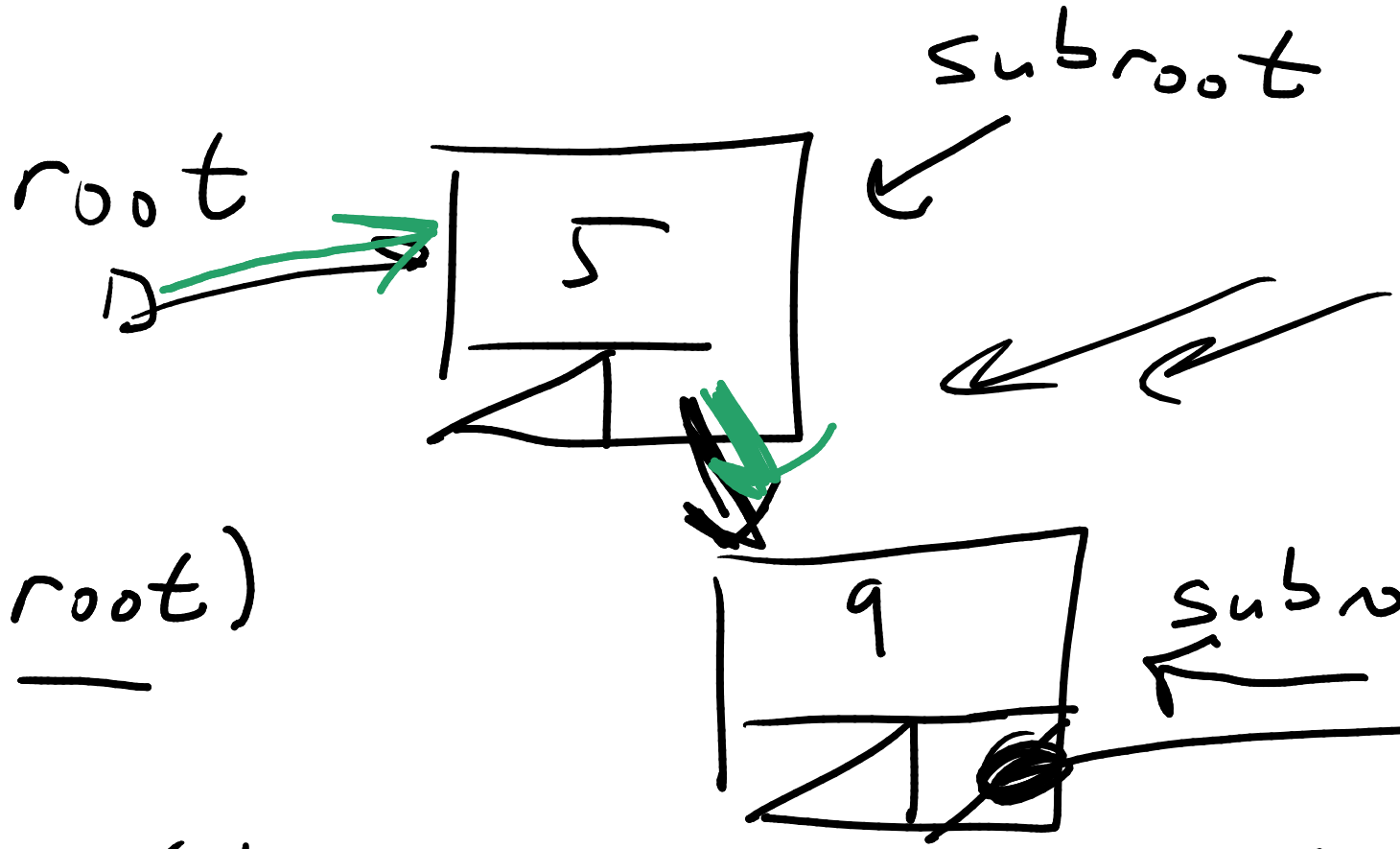



is root null? no

is 9>5? yes

subroot.right = new Node we created

12
insert




insert(12, root)

fun insert(key, subroot) {
.....
}

subroot
∅

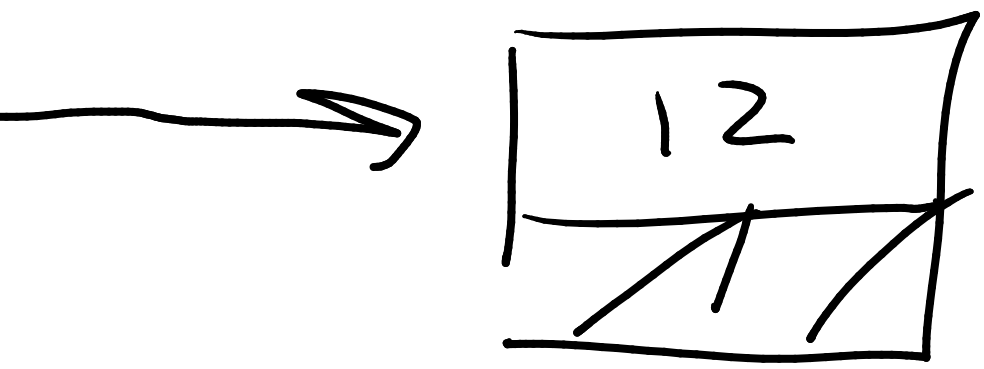
is subroot null? no
is 12 > 5? yes
subroot.right
= insert(12, subroot.right)
⇒ recursively,
is subroot null? no
is 12 > 9, yes
subroot.right =
insert(12, subroot.right)

→ another level of recursion

Is subroot null: yes

Create a new node

t



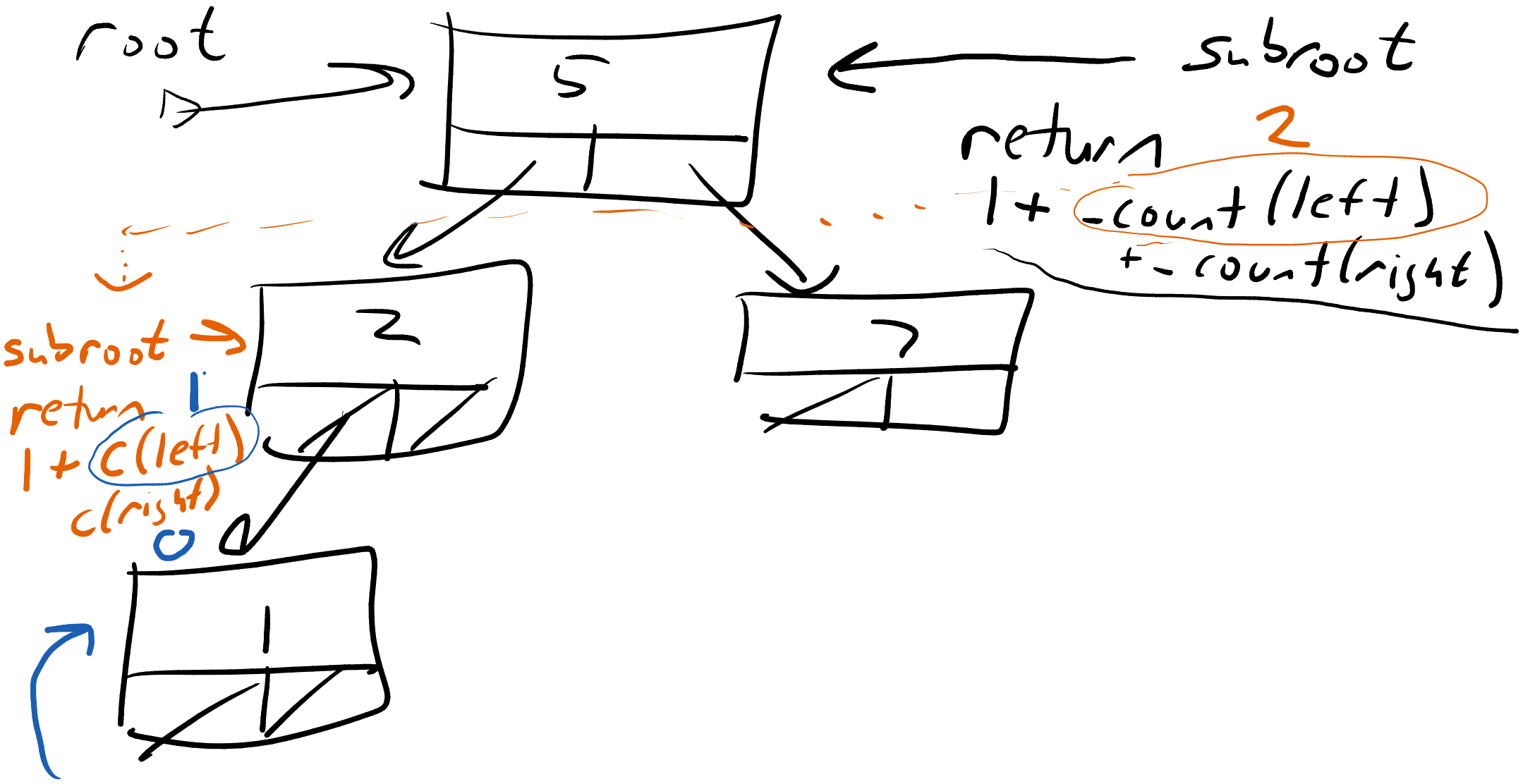

return a pointe to it

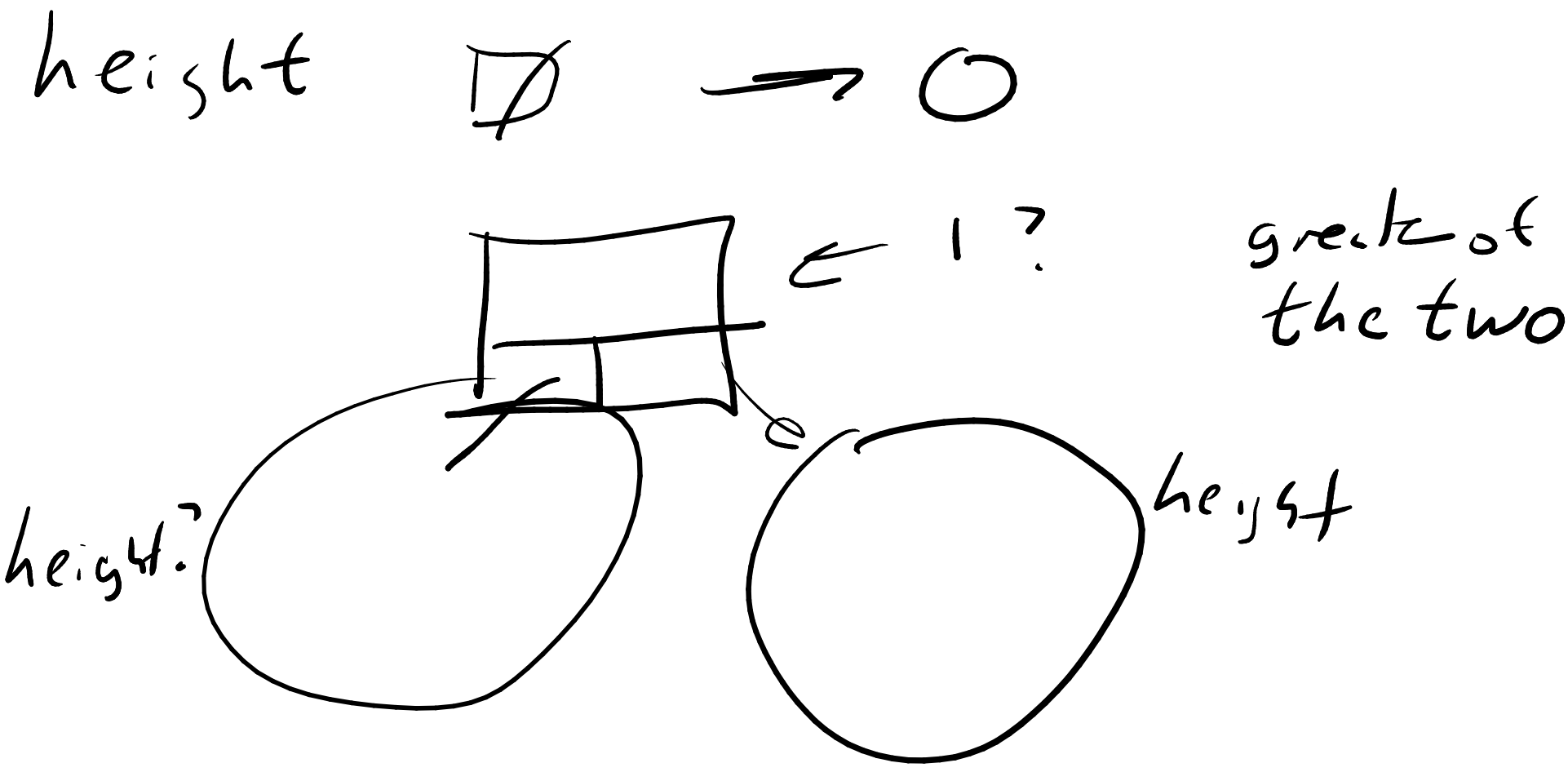
---

```
factorial {
  return 5 * factorial(n-1)
}
```

---

```
insert {
  subroot.right = insert(—,
                         subroot.right)
  return subroot
```


root
5
subroot
return 2
1 + count(left)
+ count(right)
subroot
2
7
return 1
1 + C(left)
C(right)
0
1
subroot
return 1 + count(left) + count(right)
0
0


height
1?
greater of the two
height?
height