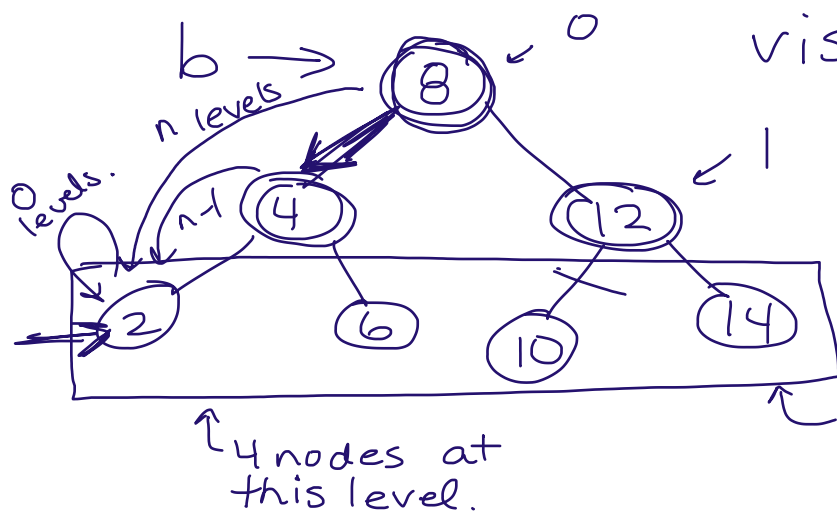


from visit_level dostring:



visit_level(b, 2, f) ^{act function}

→ visit every node in the tree rooted at b at level 2

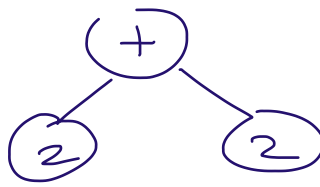
evaluation tree.

→ values for each node are either operators (+, -, *, /) or numeric operands

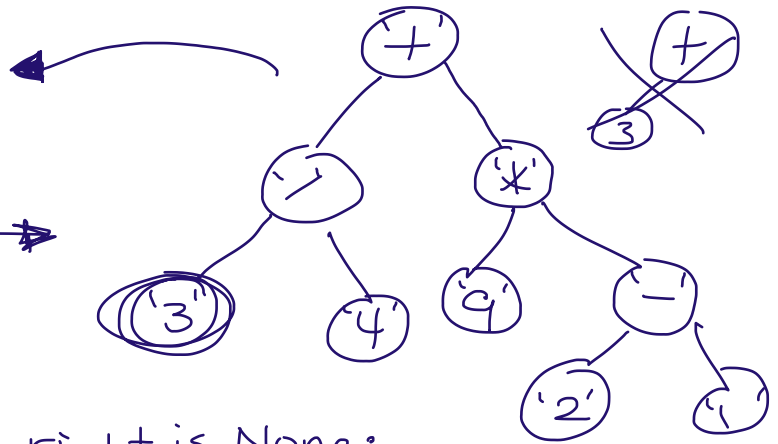
→ operators are internal nodes

→ operands are leaves

$$2 + 2 \rightarrow$$



$$(3 / 4) + (9 * (2 - 1))$$



'3' ✓
error if
b is None.

if b.left is None and b.right is None:
return eval(str(b.data))

else: # b has both children

return eval(str(evaluate(b.left))
(recursive call)

+ str(b.data)

+ str(evaluate(b.right)))

Ignoring
Order of
operations

inorder!

To make this work, needed restrictions
on tree.

==

Binary Search Tree: binary tree

where ① values are comparable to each other

② Binary Search Tree property:

- values in left subtree are
< value of root

- values in right subtree are
> value of root