Data Structures

Binary Tree Traversal 2

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

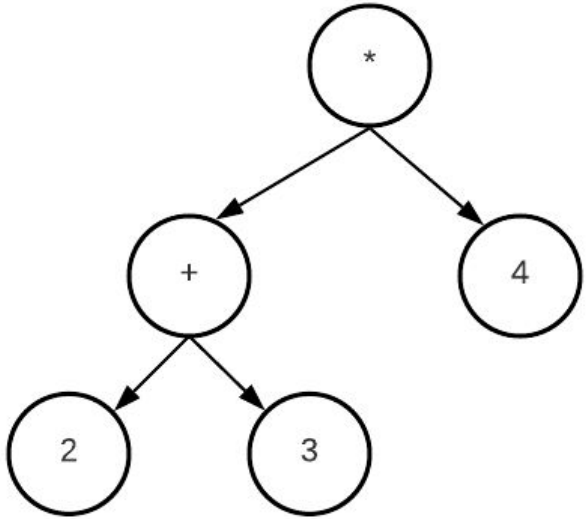
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Print Expression Tree: (2 + 3) * 4

- Starting from the top (aka, the 'current' node or 'me'), we need to think in terms of the following:
  - Print left subtree
  - Print right subtree
  - Print me

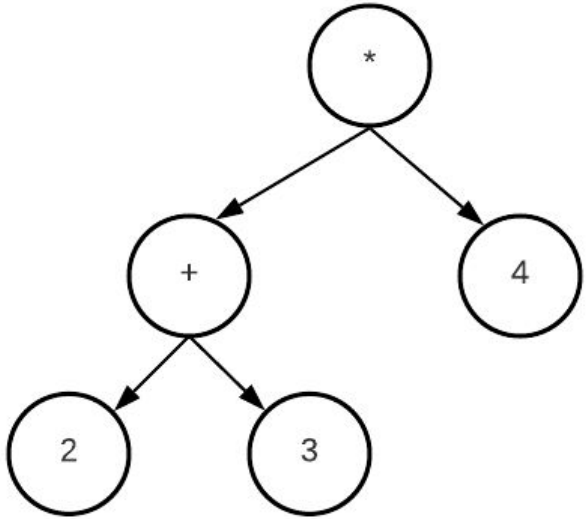
```
def print_postorder(current):
    if not current:
        return
    print_postorder(current.left)
    print_postorder(current.right)
    print(current.val, end = ' ')
```

```
print_postorder(multiply)
# 2 3 + 4 *
```



Proper Recursion Tracing

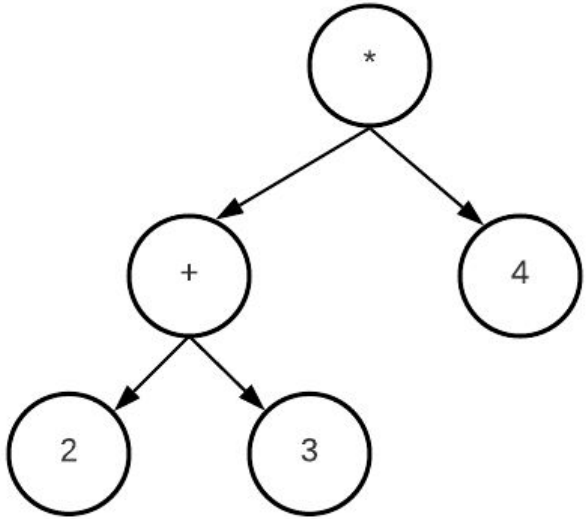
- WHAT not how!
- What is the postfix of (2 + 3) * 4?
  - 2 3 + 4 *
- What is the output of print_postorder?
  - Given an expression ⇒ prints its post-order
- What is the post-order of subtree ‘+’
  - As we did: 2 3 +
- What is the post-order of tree ‘*’
  - L = postorder(+) = 2 3 +
  - R = postorder(4) = 4
  - V = *
  - **In total: 2 3 + 4 ***



Proper Recursion Tracing

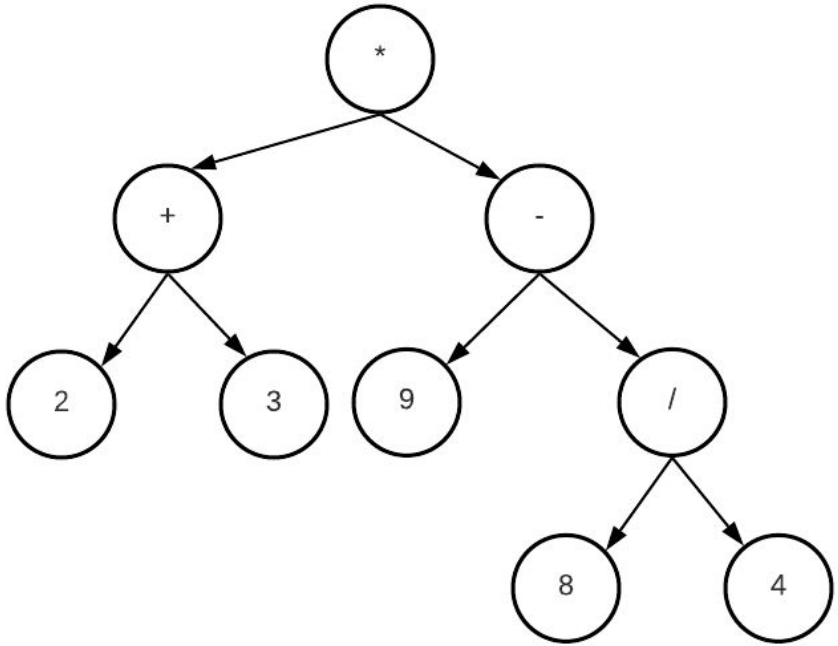
- What:
  - + subtree ⇒ 2 3 +
  - * tree ⇒ 2 3 + 4 *

```
def print_postorder(current):
    if not current:
        return
    print_postorder(current.left)
    print_postorder(current.right)
    print(current.val, end = ' ')
```



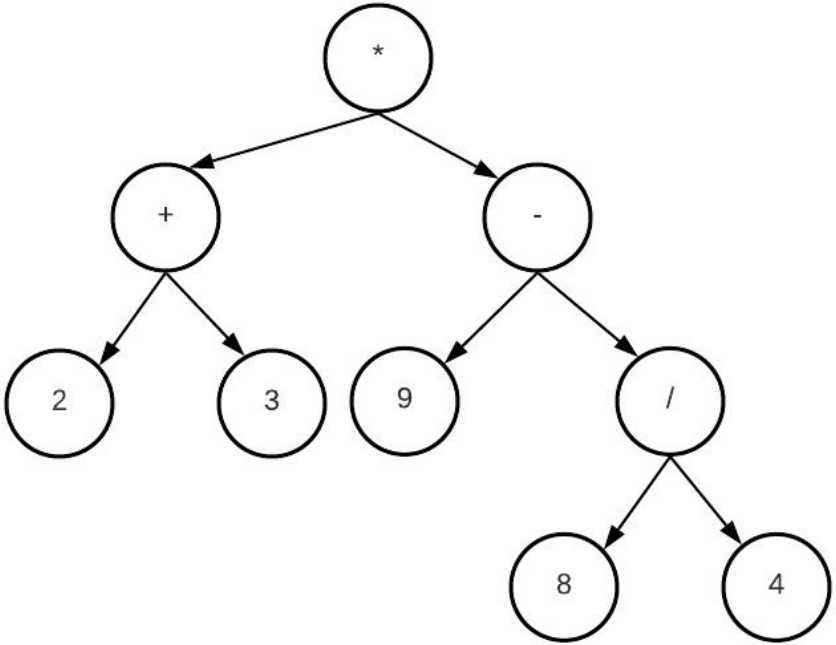
Print Expression Tree: (2 + 3) * (9 - 8 / 4)

```
plus = Node('+')
plus.left = Node('2')
plus.right = Node('3')

div = Node('/')
div.left = Node('8')
div.right = Node('4')

minus = Node('-')
minus.left = Node('9')
minus.right = div

multiply = Node('*')
multiply.left = plus
multiply.right = minus

print_postorder(multiply)
```



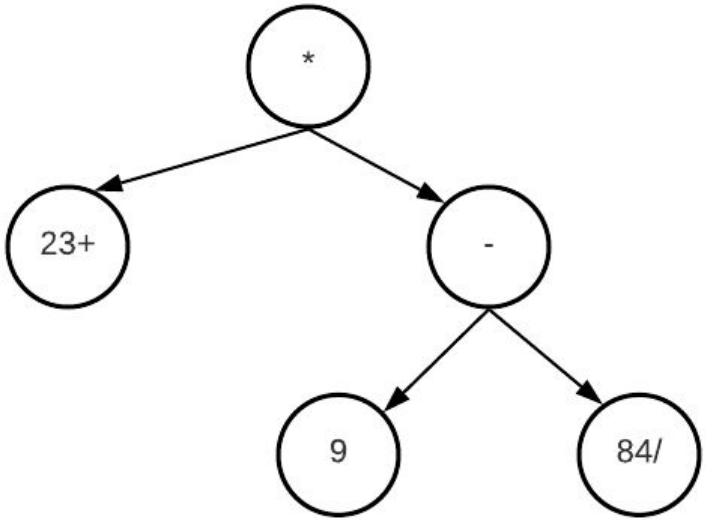
Print Expression Tree: (2 + 3) * (9 - 8 / 4)

- What is the postfix expression for:
- + subtree? 2 3 +
- / subtree? 8 4 /



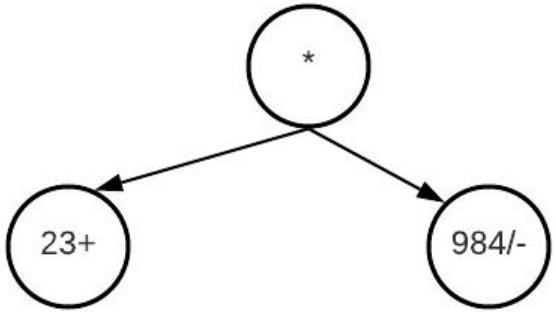
Print Expression Tree: (2 + 3) * (9 - 8 / 4)

- - subtree?
  - Left = 9
  - Right = 8 4 /
  - Value = -
  - Total: 9 **8 4 /** -



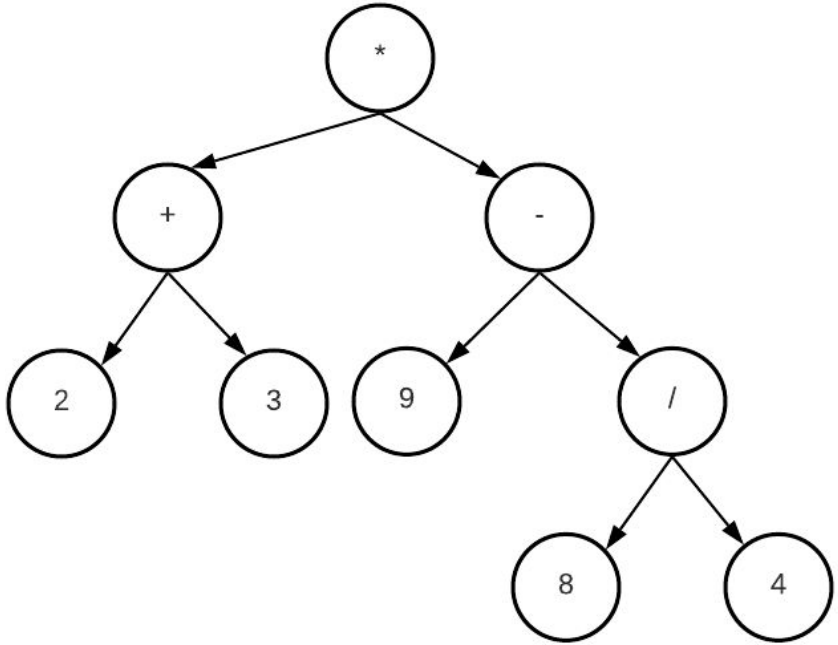
Print Expression Tree: (2 + 3) * (9 - 8 / 4)

- * subtree?
  - Left = 2 3 +
  - Right = 8 4 / -
  - Value = *
  - Total: 2 3 + 9 8 4 / - *



Print Expression Tree: (2 + 3) * (9 - 8 / 4)

- What:
  - + subtree ⇒ 2 3 +
  - / subtree ⇒ 8 4 /
  - - subtree ⇒ 9 8 4 / -
  - * subtree ⇒ 2 3 + 9 8 4 / - *




```
def print_postorder(current):
    if not current:
        return
    print_postorder(current.left)
    print_postorder(current.right)
    print(current.val, end = ' ')
```

Your turn

- Trace it out and ensure you fully understand it!

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."