LEC 12

CSE 123

Binary Trees

Questions during Class?
Raise hand or send here

sli.do #cse123

BEFORE WE START

Talk to your neighbors:

What is your favorite Thanksgiving food?
(or favorite holiday-specific food from your background)

Respond on sli.do!

Instructor: Nathan Brunelle

TAs:

Cora	Chris	Crystal	Anya	Evan
Jonah	Nhan	Saachi	Ishita	Sahana
Chloe	Shiven	Nate	Renee	Yuntong
Amiya	Shashmit	Akshaya	Vihan	Shreedhaarini
Roshan	Adiyan	Nancy	Rachel	Ananya
Ryuta	Mimi	Isaac	Megan	

Lecture Outline: Announcements

- Announcements
- Binary Tree Review
- Traversals
- Practice!

Announcements

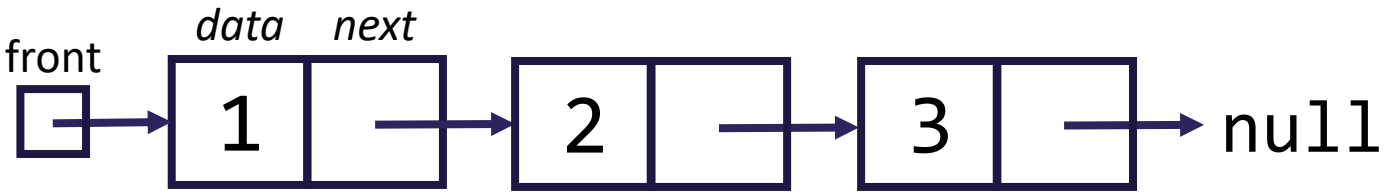
- PQ1 in section on Tuesday 11/17
  - Practice quiz is in Ed resources
- P2 due Wednesday 11/18 @ 11:59pm
- CQ2 begins Thursday 11/19
  - Canvas sign ups will be available this evening
  - Do not sign up until I post an Ed announcement inviting you to do so

Lecture Outline: Binary Tree Review

- Announcements
- **Binary Tree Review**
- Traversals
- Practice!

Binary Trees: compared to Linked Lists

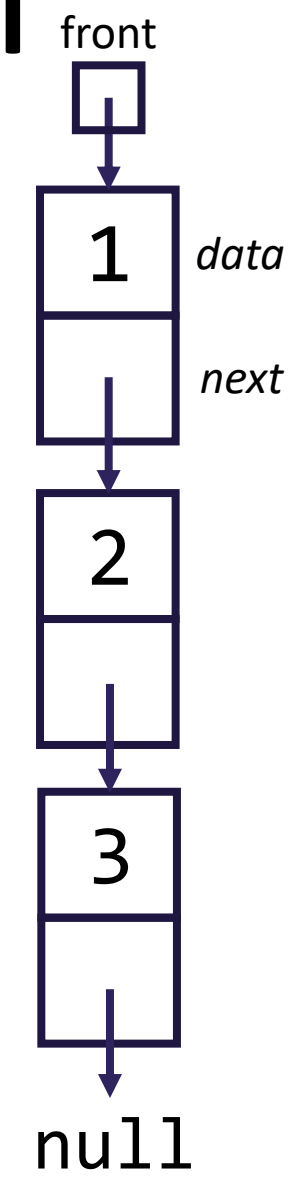
- Last data structure of the quarter!
  - Very similar to `LinkedLists`…





Binary Trees: Linked Lists vertical

- Last data structure of the quarter!
  - Very similar to `LinkedLists`…

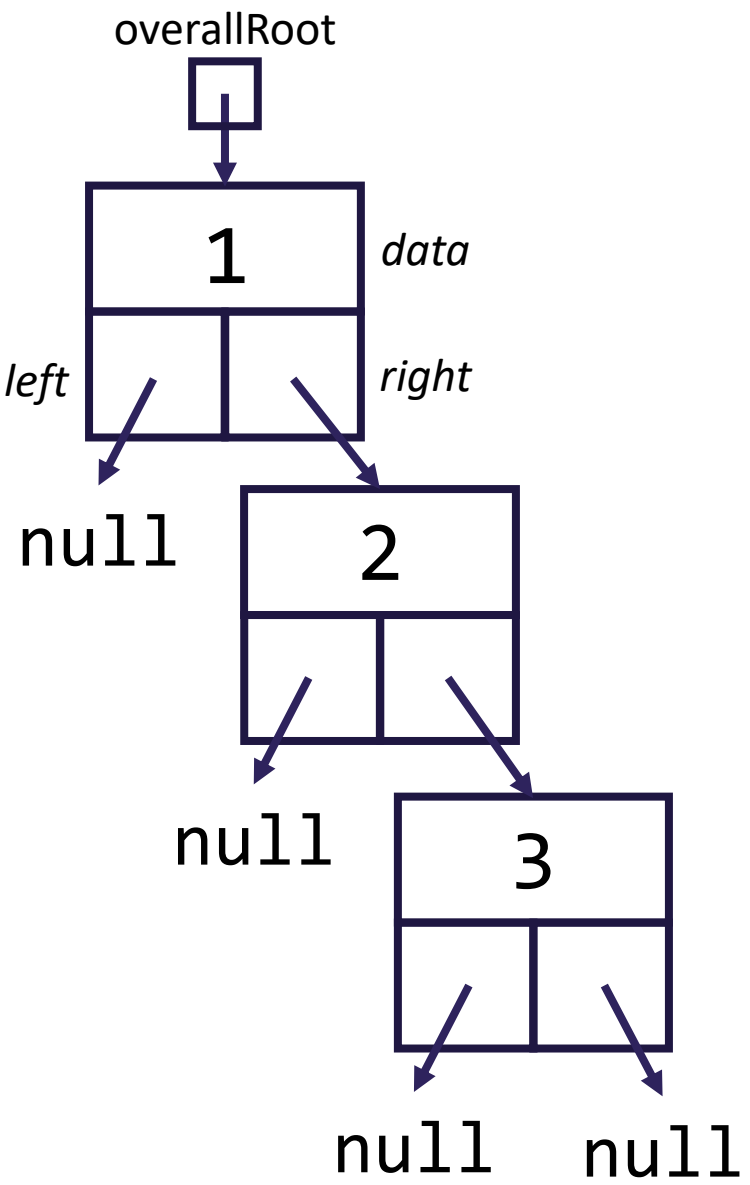




Binary Trees: Tree Structure

- Last data structure of the quarter!
  - Very similar to `LinkedLists`…

- Linked `TreeNodes` w/ 3 fields:
  - int data, `TreeNode` left, `TreeNode` right
  - Doubly complicated!

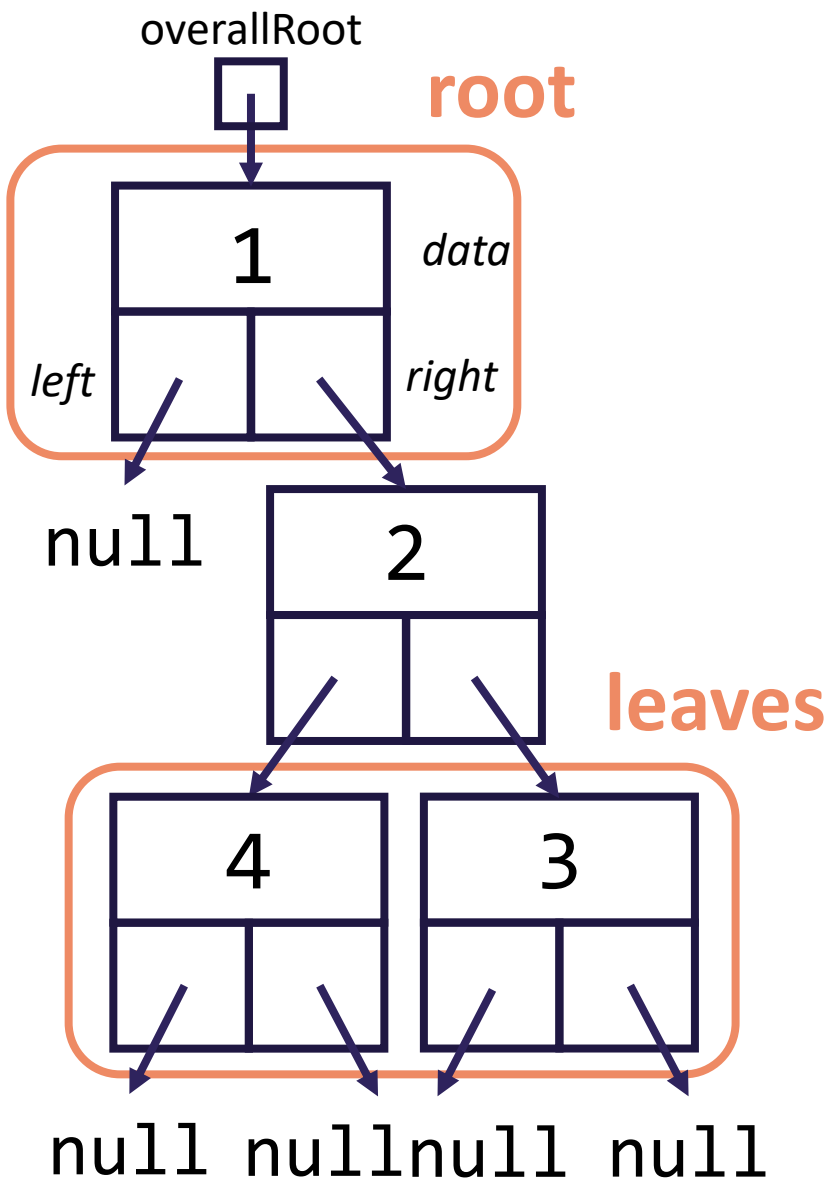




Binary Trees: root and leaves

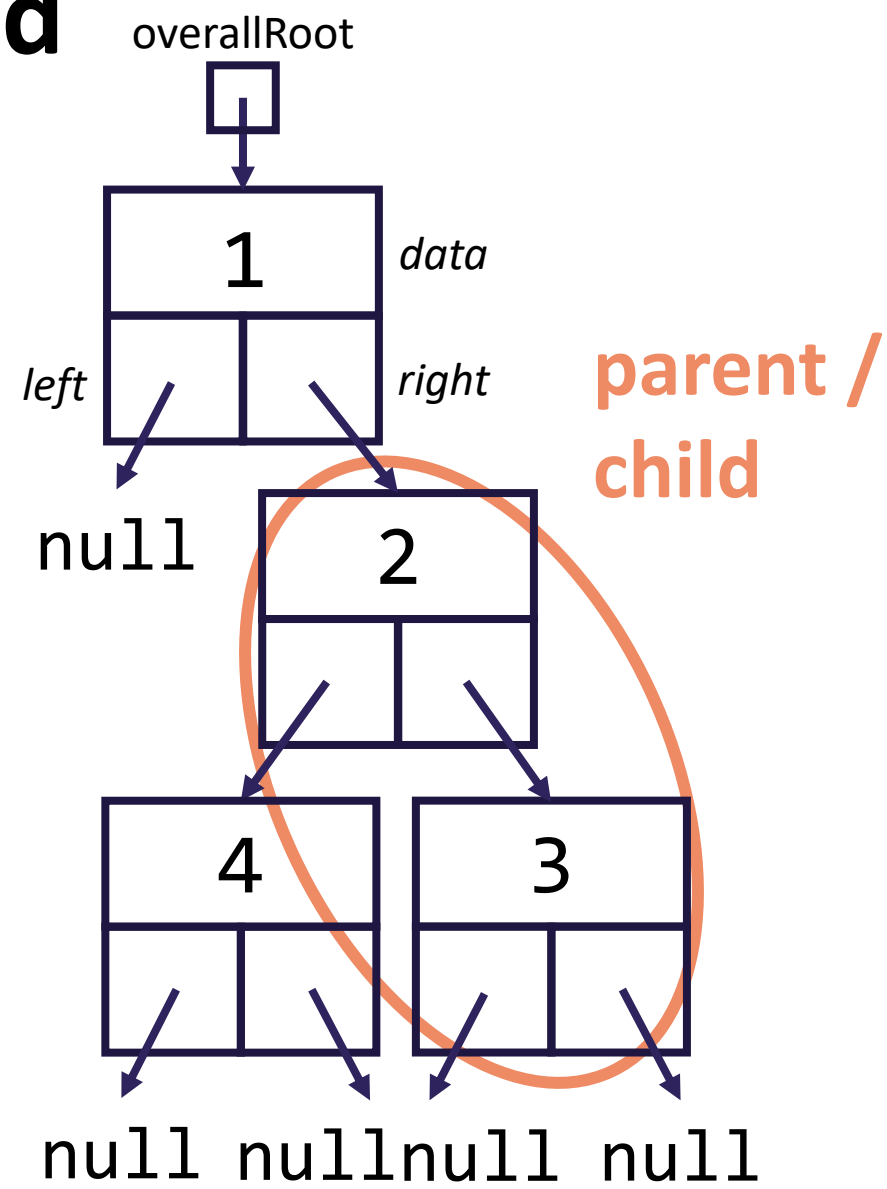
- Last data structure of the quarter!
  - Very similar to `LinkedLists`…
- Linked `TreeNodes` w/ 3 fields:
  - int data, `TreeNode` left, `TreeNode` right
  - Doubly complicated!
- Similar to trees?
  - Close enough!
  - Terminology: root / leaves
- Other terminology as well





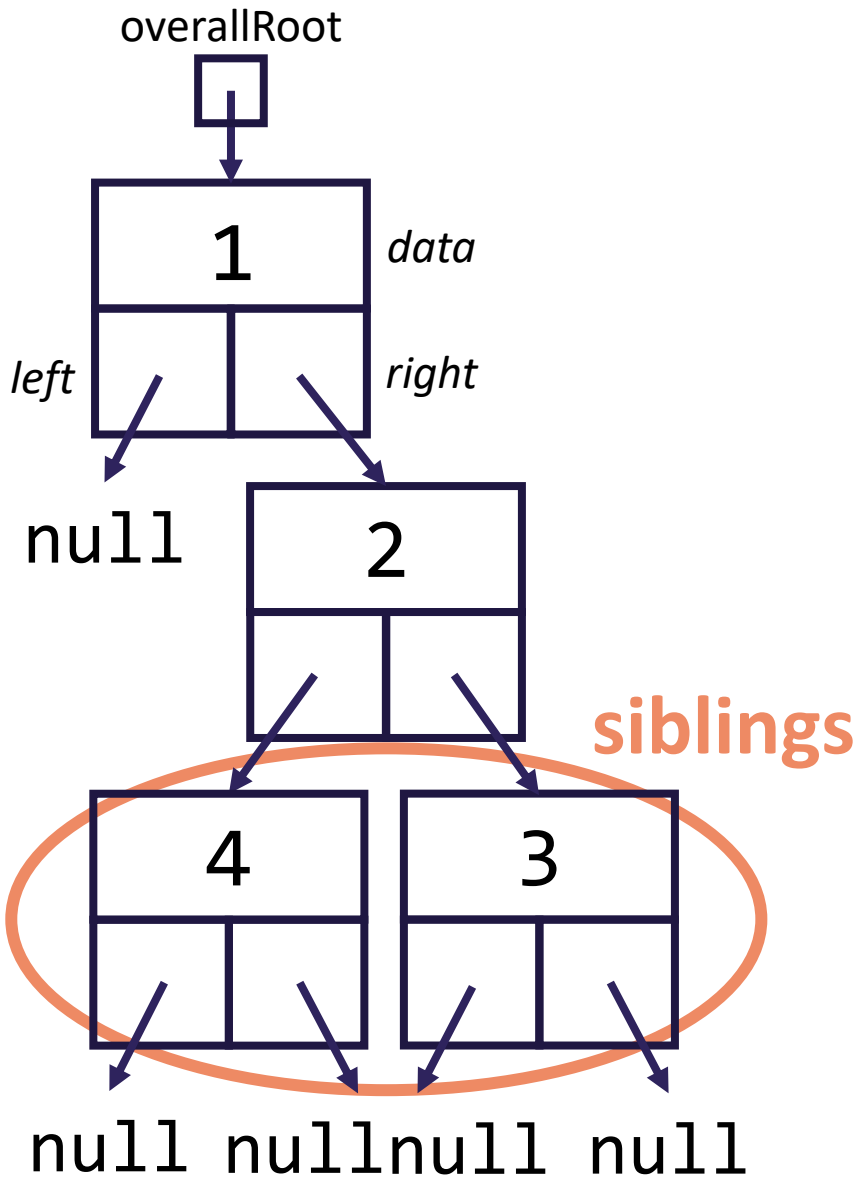
Tree Terminology: parent/child





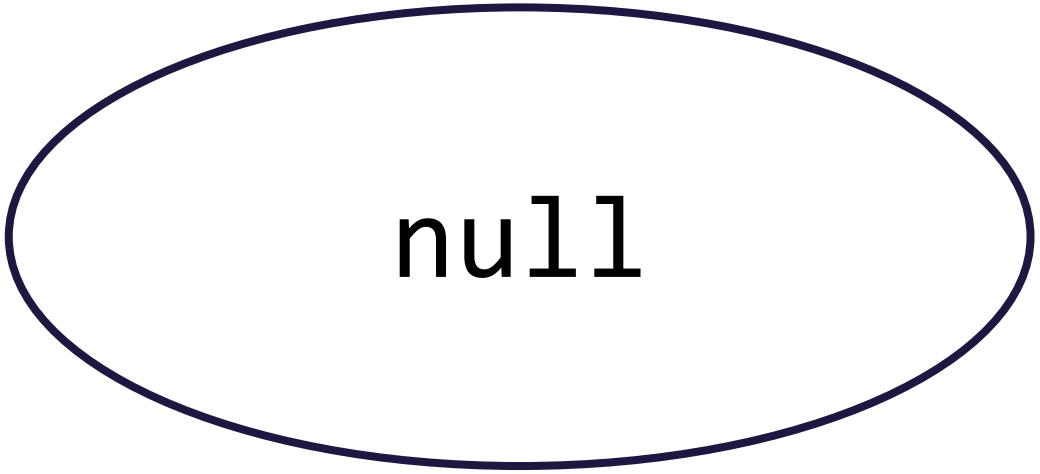
Tree Terminology: siblings



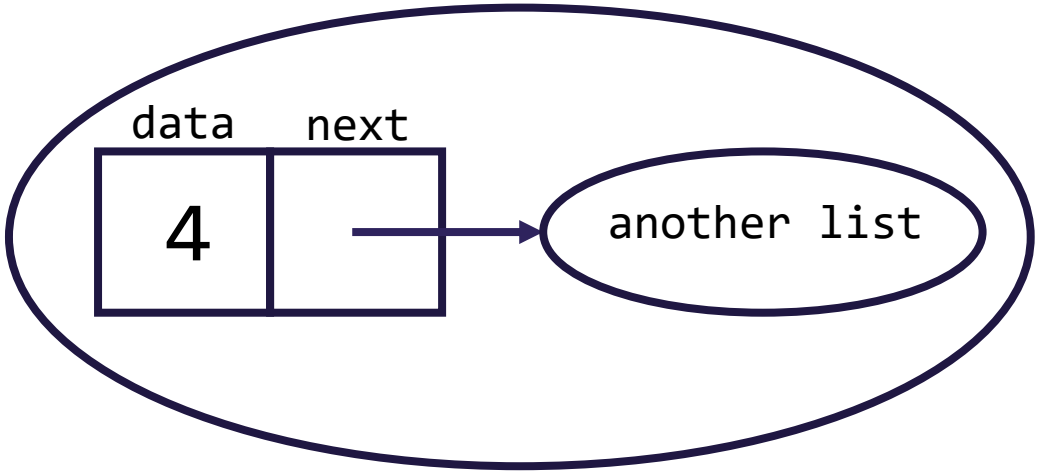


Linked Lists [Review]

- A linked list is either:



Empty list



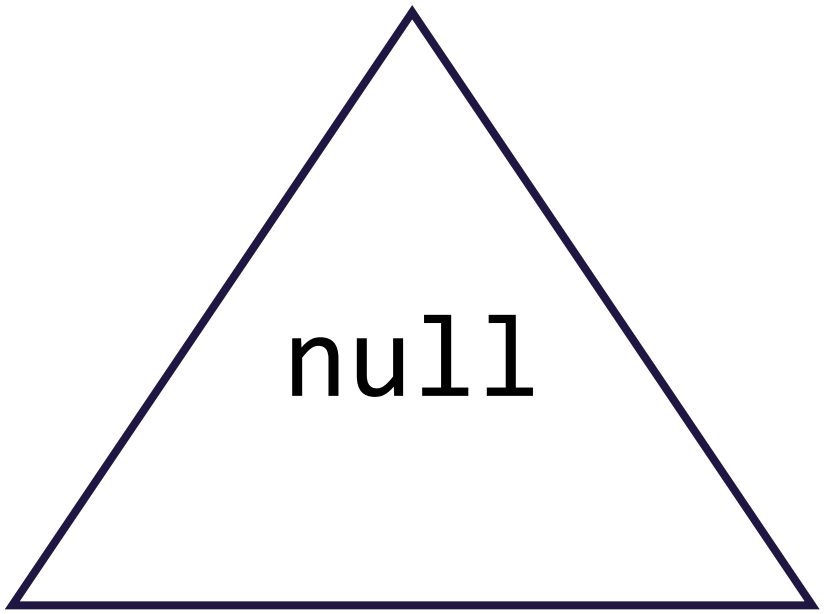
Node w/ another linked list

This is a recursive definition!

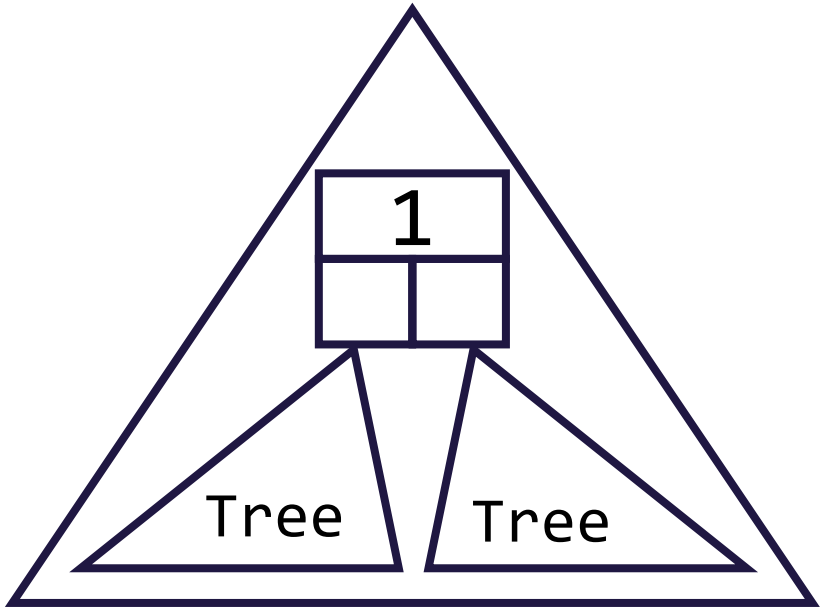
A list is either empty or a node with another list!

Binary Trees: Recursive Definition

- A Binary Tree is either:



Empty tree



Node w/ two subtrees

This is a recursive definition!
A tree is either empty or a node with two more trees!

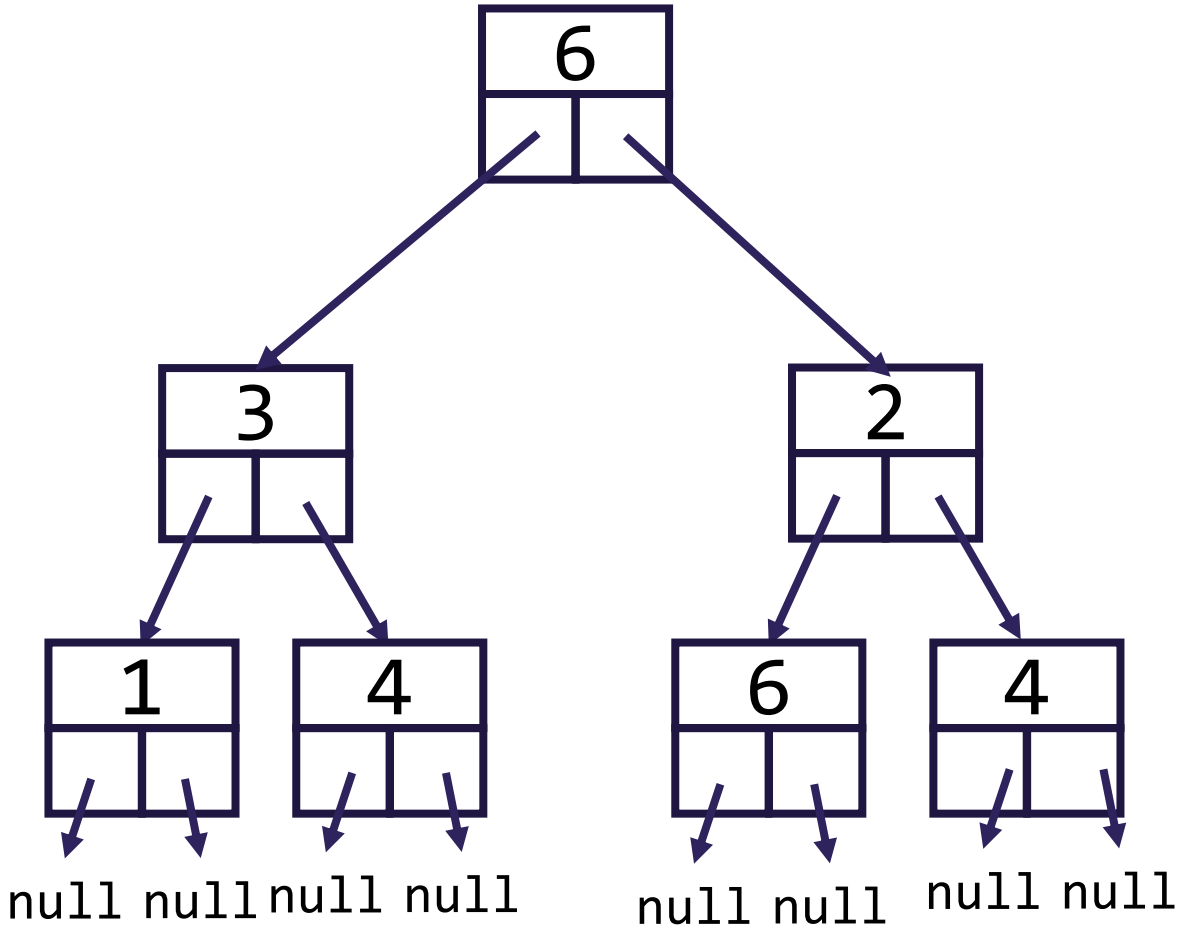
Binary Tree Programming

- Programs look very similar to Recursive LinkedList!

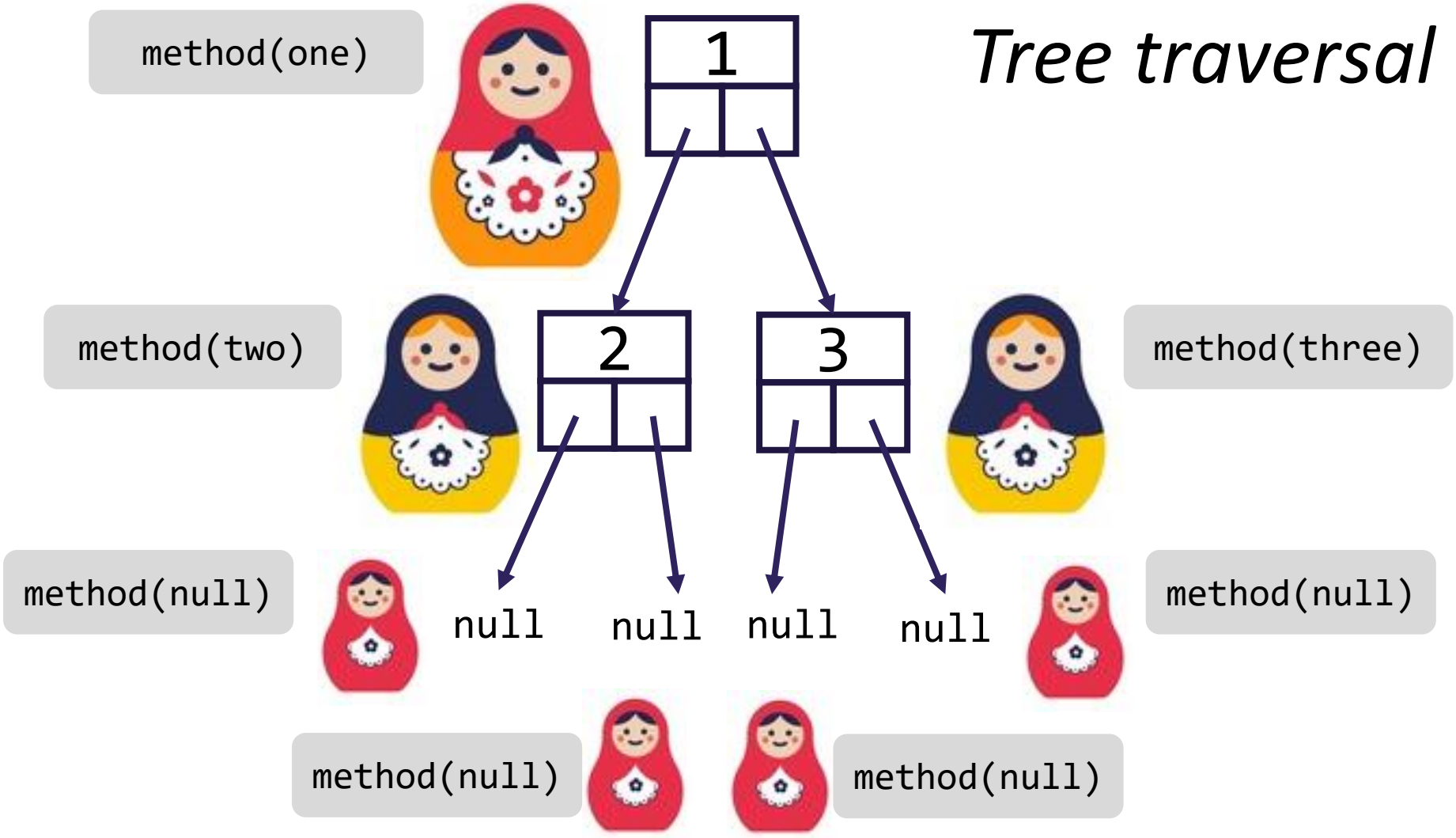
- Guaranteed base case: empty tree
  - Simplest possible input, should immediately know the return
- Guaranteed public / private pair
  - Need to know which subtree you're currently processing
- If modifying, we use `x = change(x)`
  - Don't stop early, return updated subtree (`IntTreeNode`)

- Let's trace through an example together…

How would I print this tree?



Tracing Through Binary Tree Programming

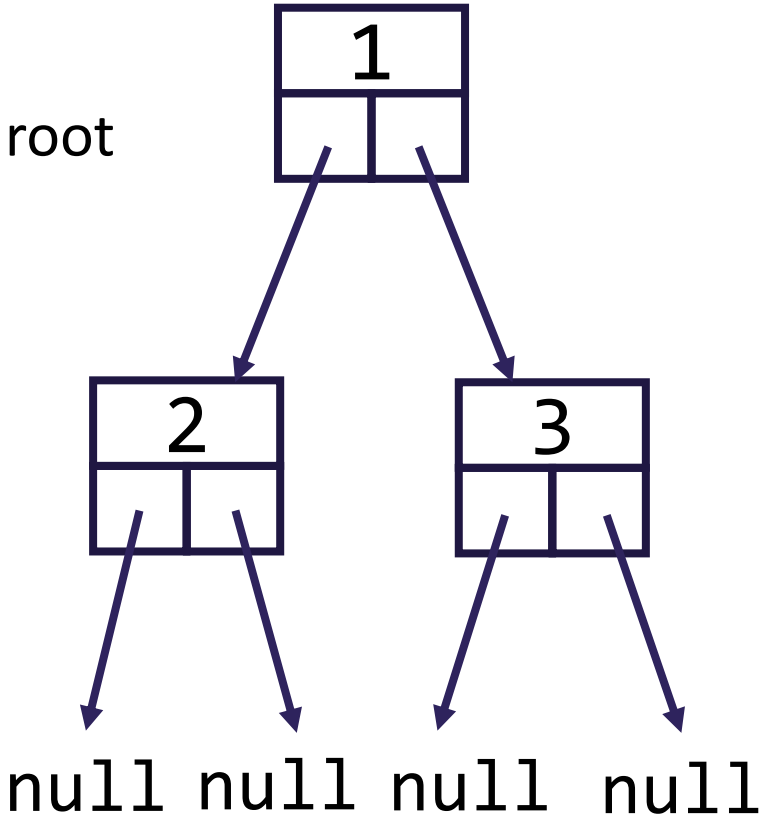


Lecture Outline: Traversals

- Announcements
- Binary Tree Review
- **Traversals** 
- Practice!

Binary Tree Traversals

- 3 different primary traversals
  - All concerned with when you process your current root



- Pre-order traversal:
  - Process **root**, left subtree, right subtree
- In-order traversal:
  - Process left subtree, **root**, right subtree
- Post-order traversal:
  - Process left subtree, right subtree, **root**

Sometimes different traversals yield different results

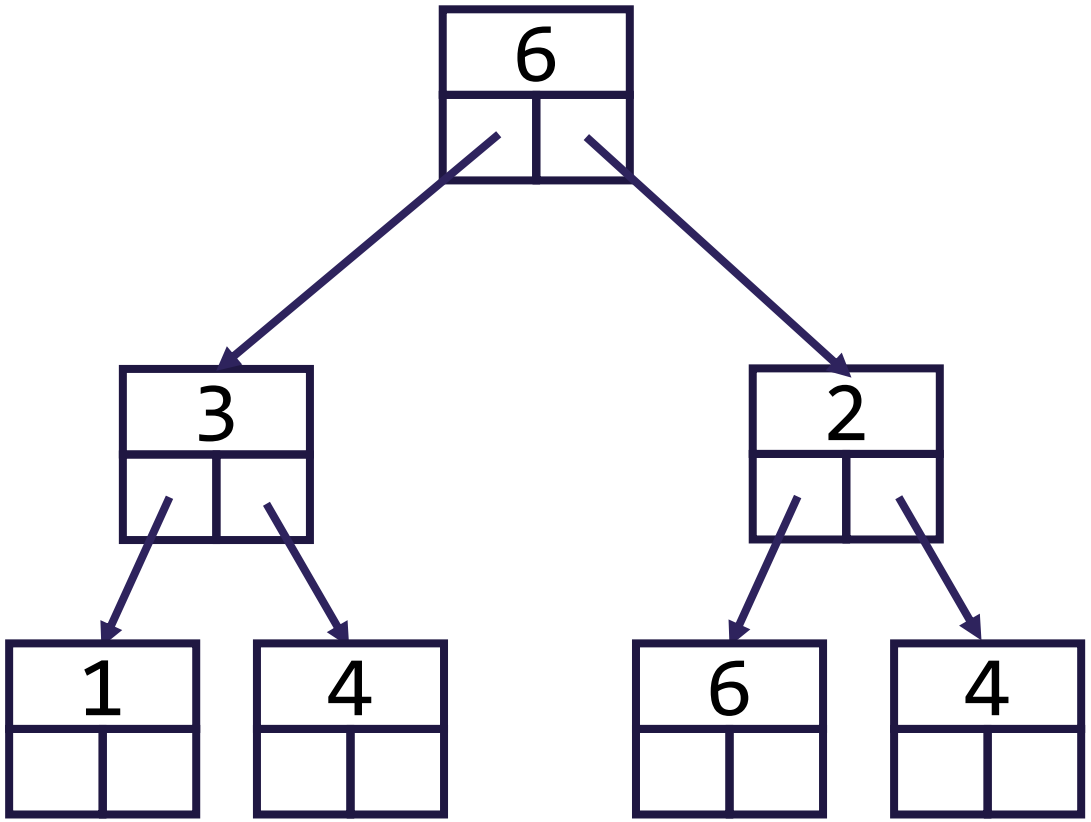


Practice : Think



sli.do #cse123

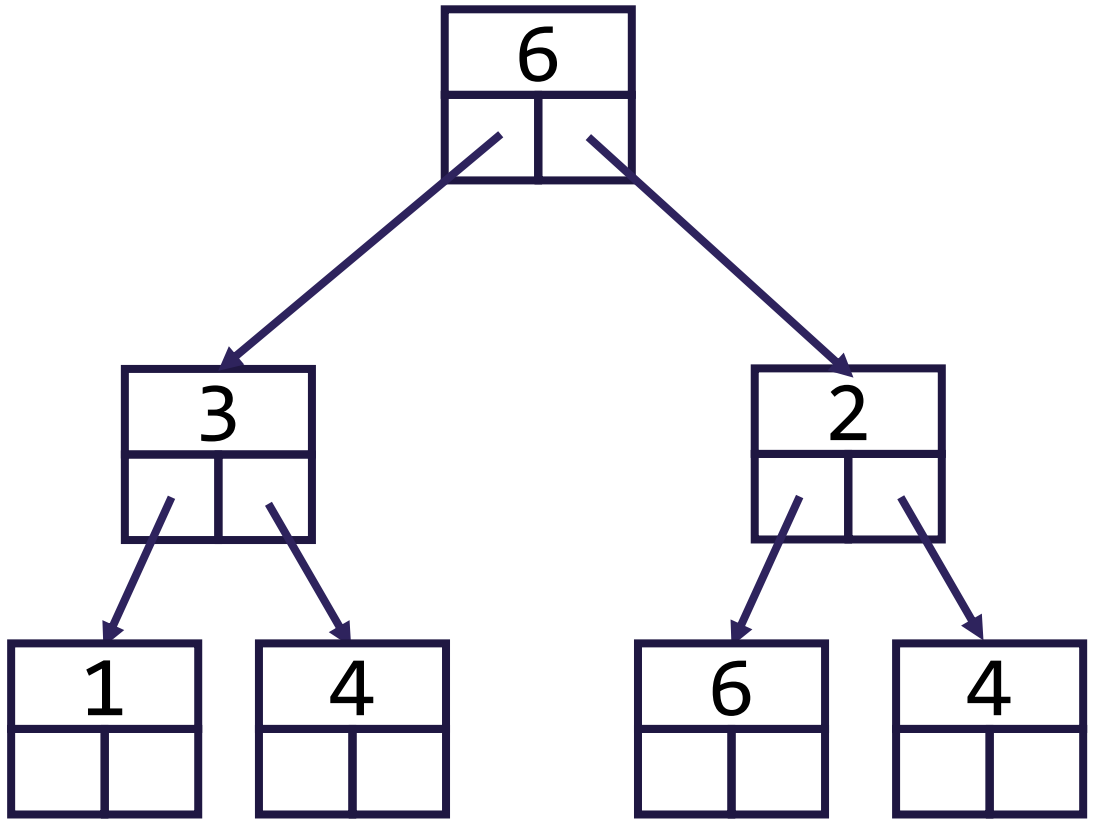
Enter the order in which the nodes of this tree would be visited in a pre-order traversal.



Practice : Pair

sli.do #cse123

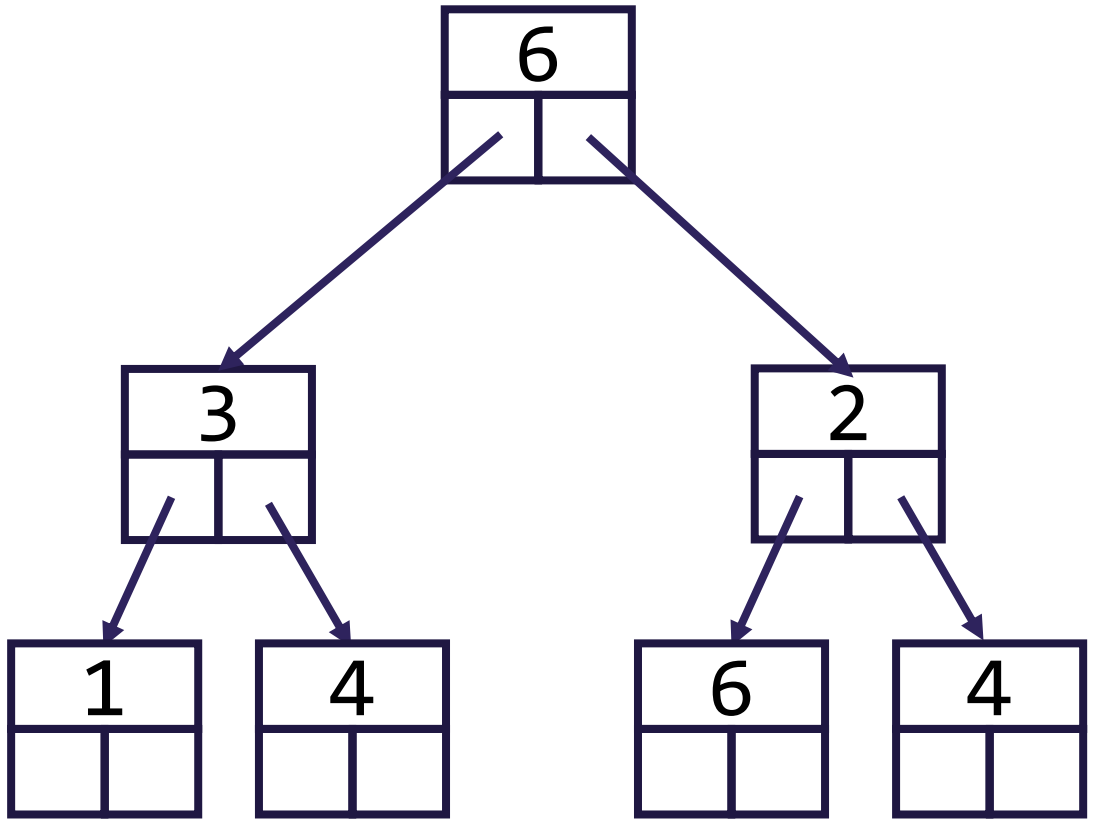
Enter the order in which the nodes of this tree would be visited in a pre-order traversal.



Practice : Pair

sli.do #cse123

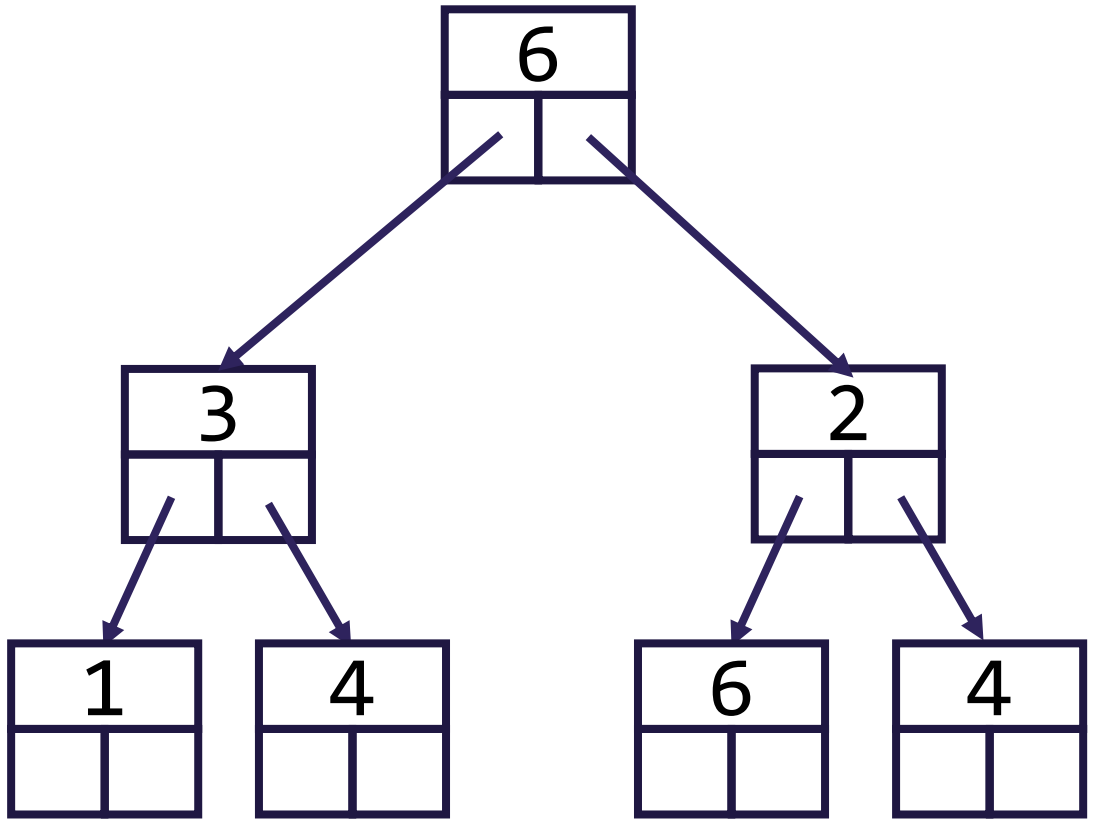
Enter the order in which the nodes of this tree would be visited in a pre-order traversal.



Practice : Pair

sli.do #cse123

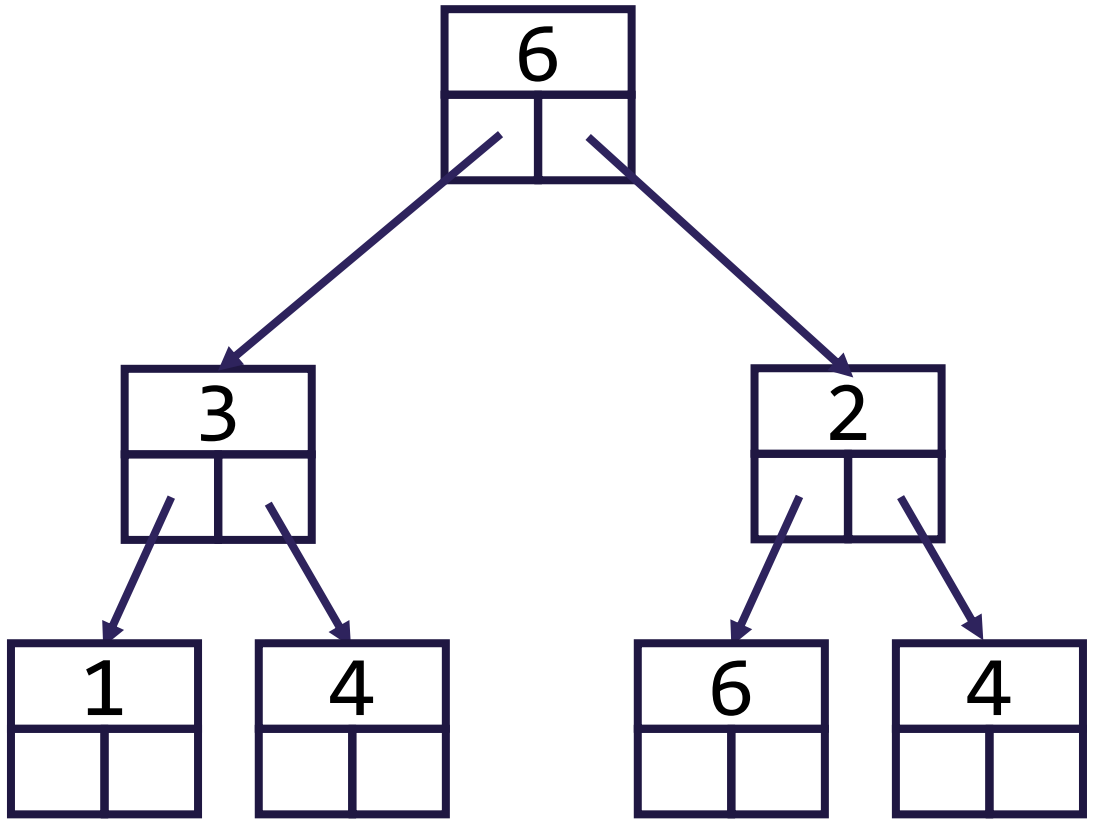
Enter the order in which the nodes of this tree would be visited in a post-order traversal.



Lecture Outline: Practice!

- Announcements
- Binary Tree Review
- Traversals
- **Practice!** 

Tracing through size




```
public int size() {
    return size(overallRoot);
}

private int size(IntTreeNode currentRoot) {
    if (currentRoot == null) {
        return 0;
    } else {
        return 1 +
               size(currentRoot.left) +
               size(currentRoot.right);
    }
}
```