LEC 13

CSE 123

Binary Trees

BEFORE WE START

Talk to your neighbors:

What's your favorite tree?

Instructor: James Wilcox

Questions during Class?

Raise hand or send here

sli.do #cse123

Announcements

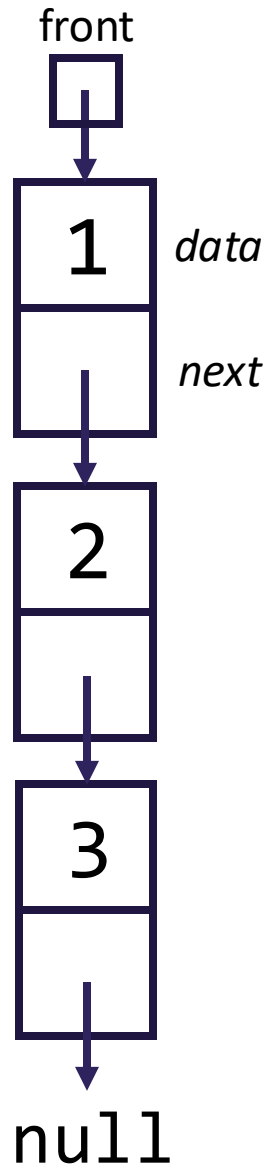
- R5 due tonight
- P2 due Wednesday (11/13)
- Quiz 1 grades out early next week
- Monday is a university holiday

Binary Trees

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

Binary Trees

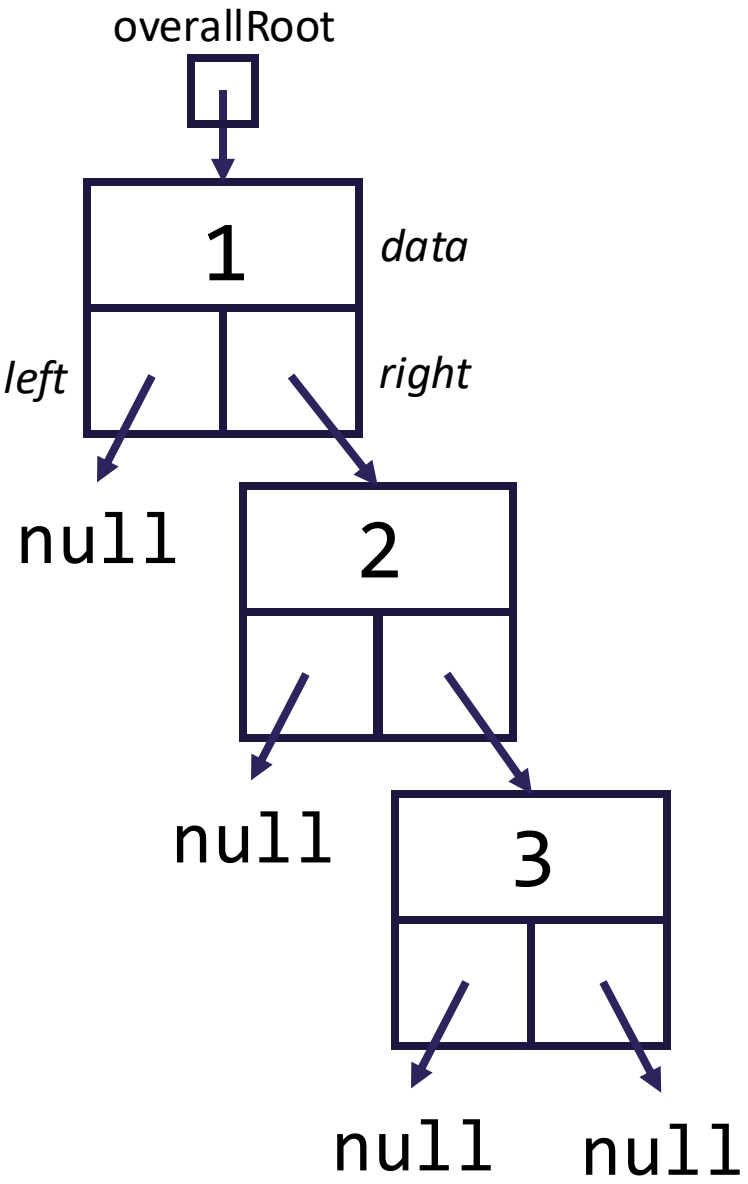
- Last data structure of the quarter!
  - Very similar to `LinkedLists`…



Binary Trees

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

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

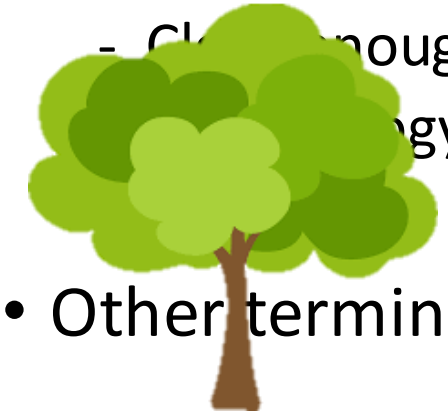


Binary Trees

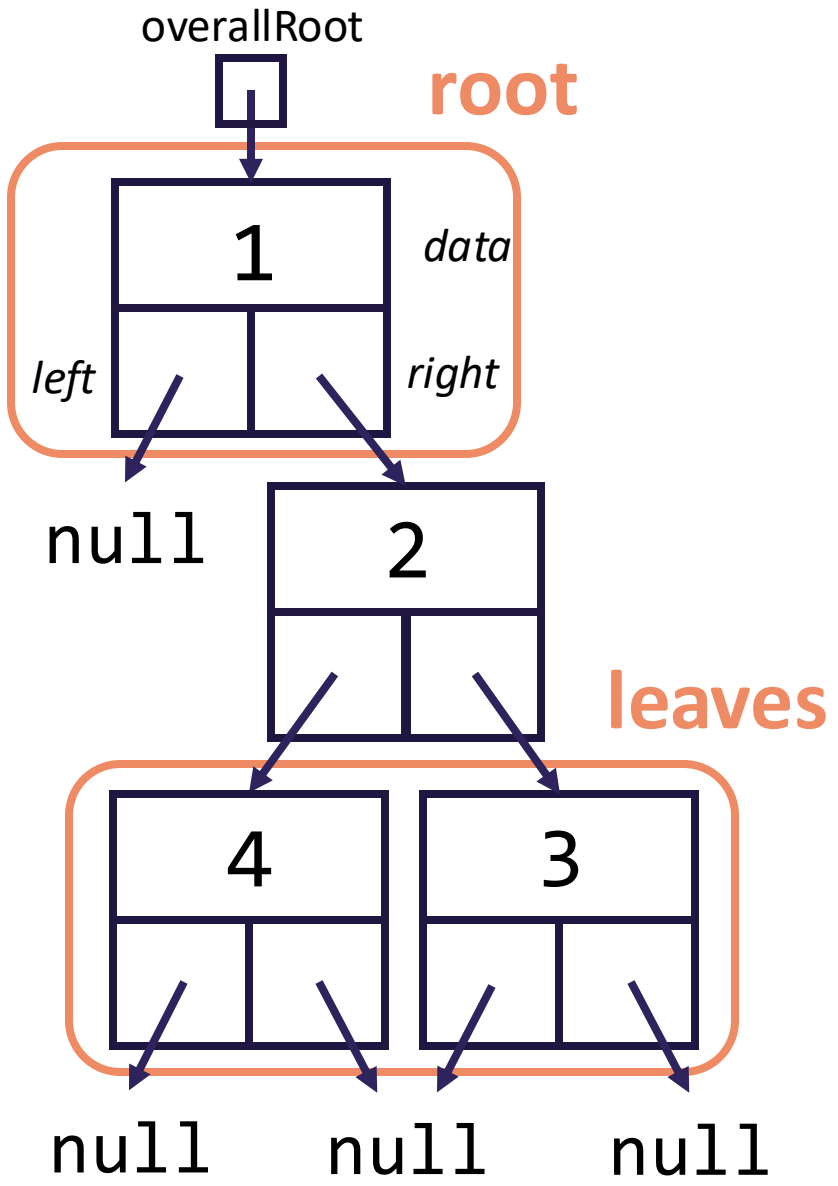
- Last data structure of the quarter!
  - Very similar to `LinkedLists`…

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

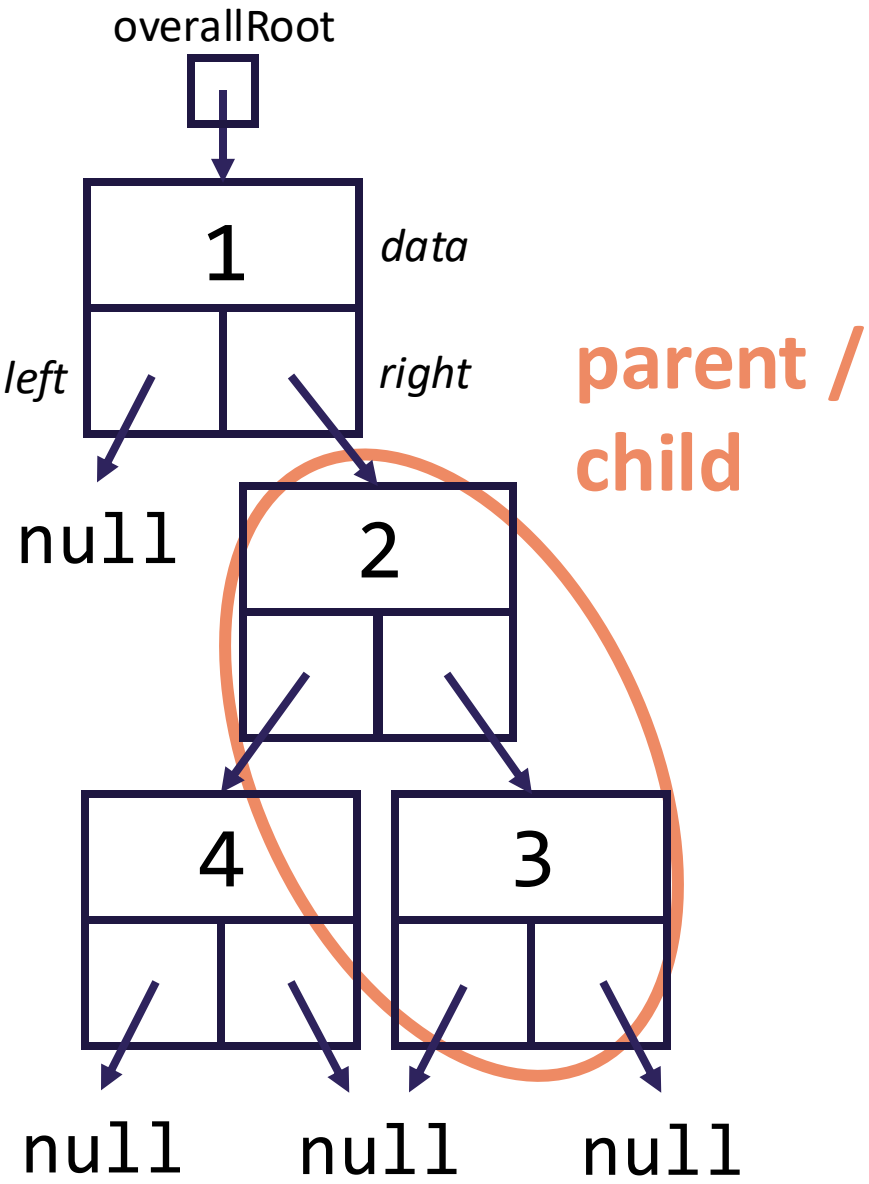
- Similar to trees?
  - Cl[illegible]nough!
  - [illegible]gy: root / leaves



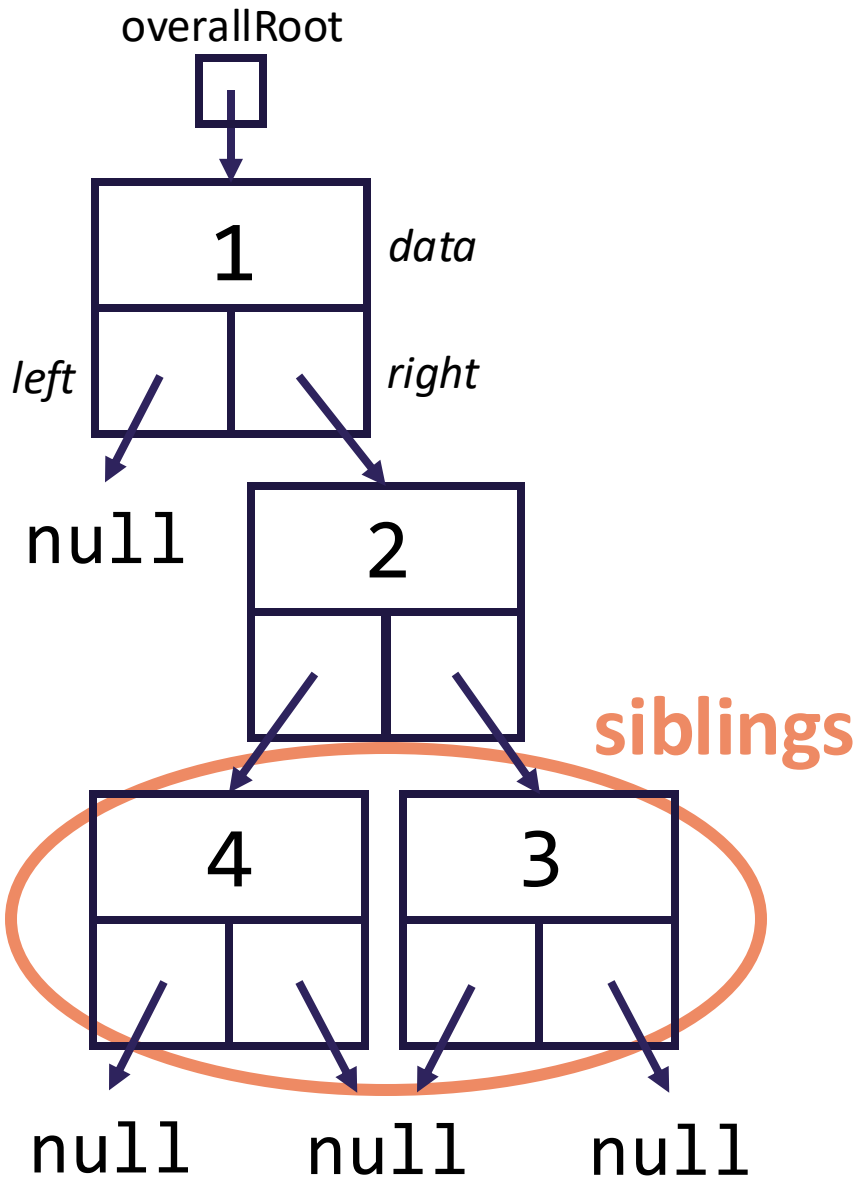
- Other terminology as well



Tree Terminology

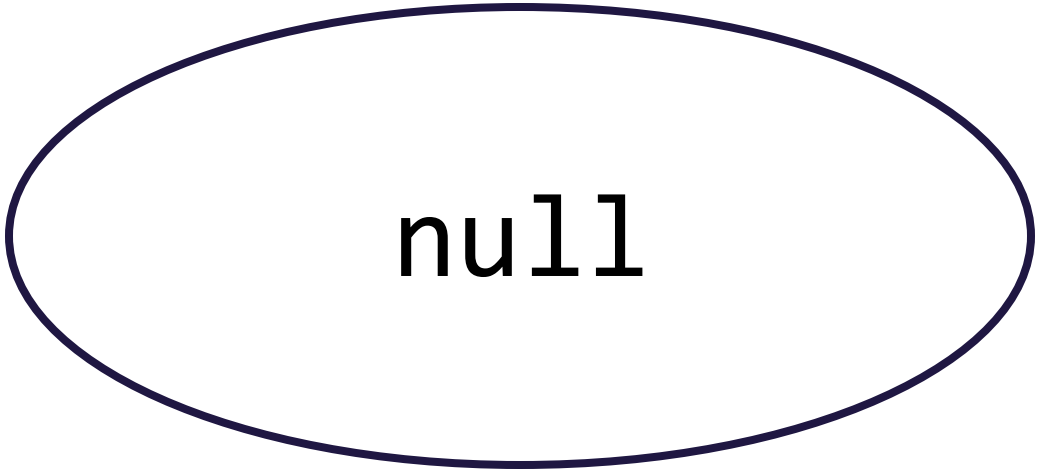


Tree Terminology

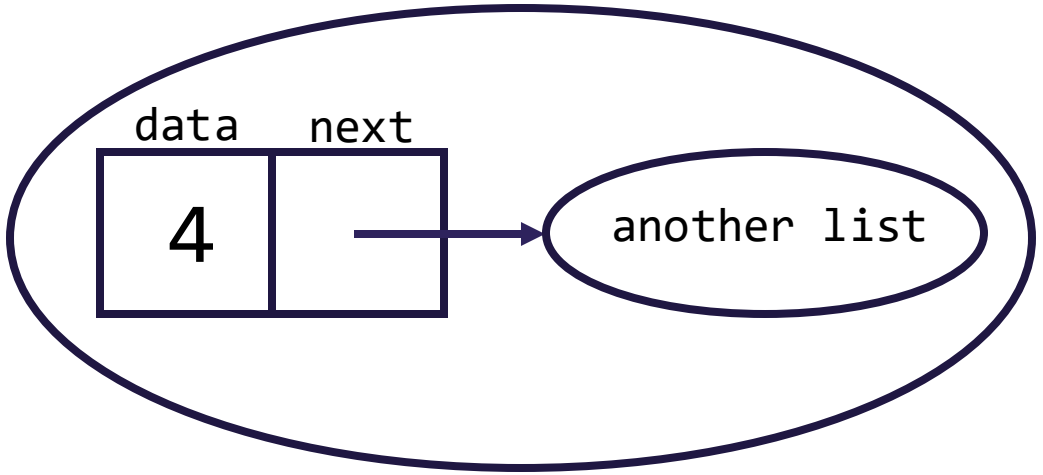


Linked Lists [Review]

- A linked list is either:



Empty list



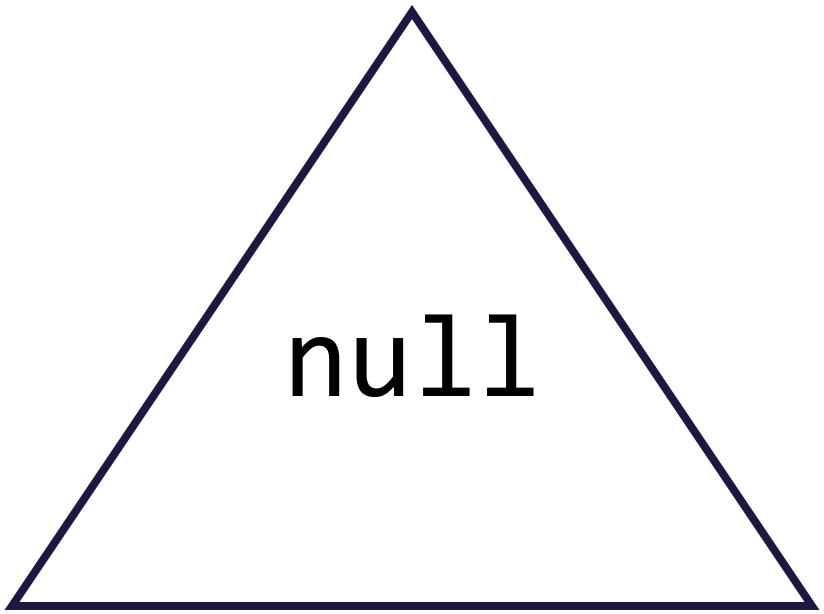
Node w/ another linked list

This is a recursive definition!

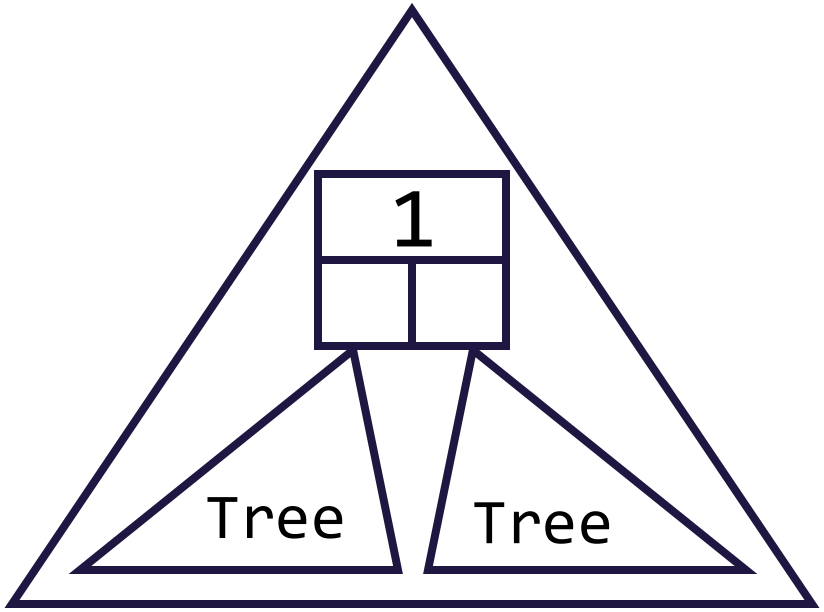
A list is either empty or a node with another list!

Binary Trees

- 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!

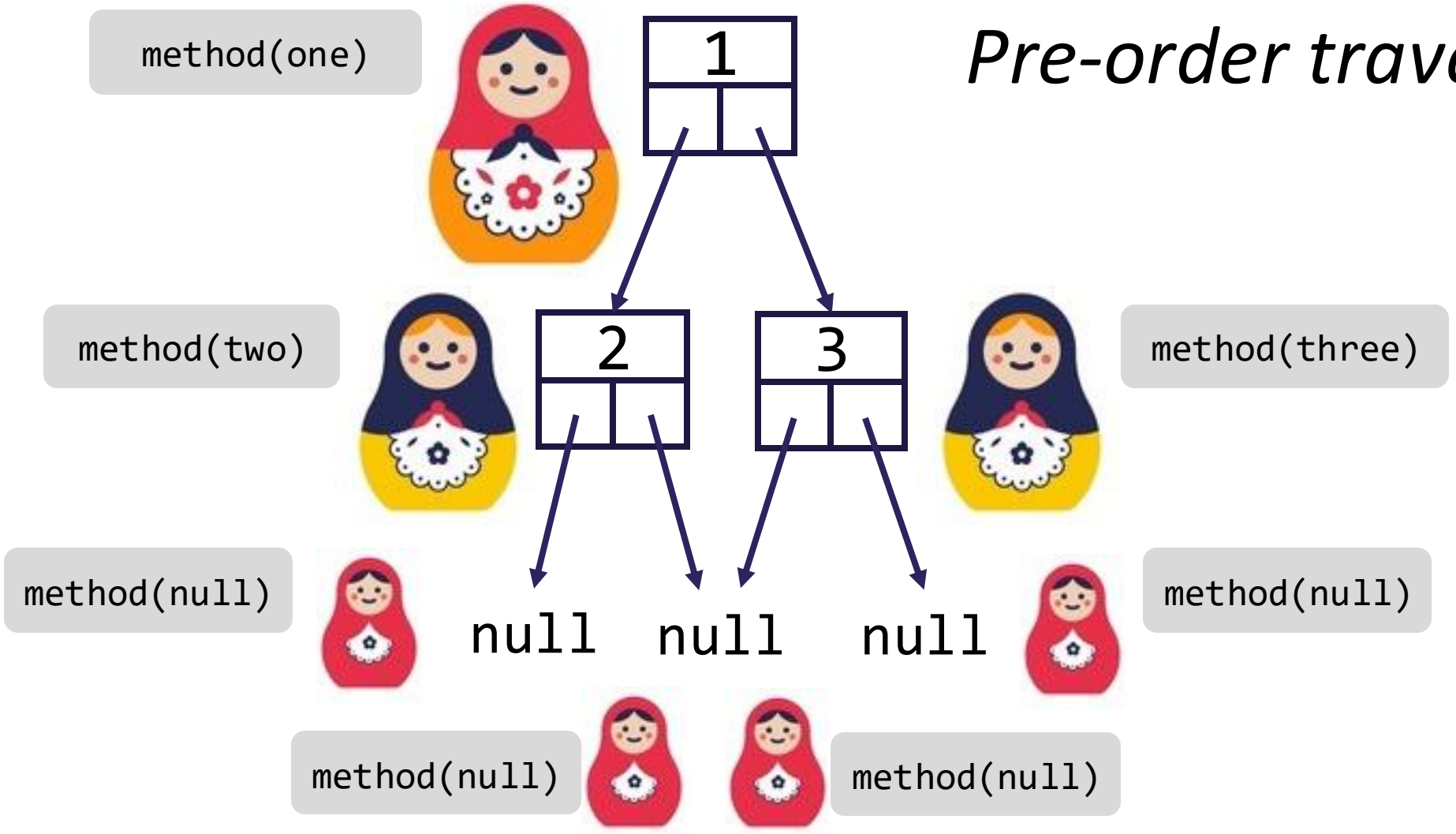
Binary Tree Programming

- 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…

Binary Tree Programming



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