

LEC 14

CSE 123

Binary Trees

BEFORE WE START

*Talk to your neighbors:**How did Quiz 1 go?*

Instructor: James Wilcox

Questions during Class?
Raise hand or send here

sli.do #cse123

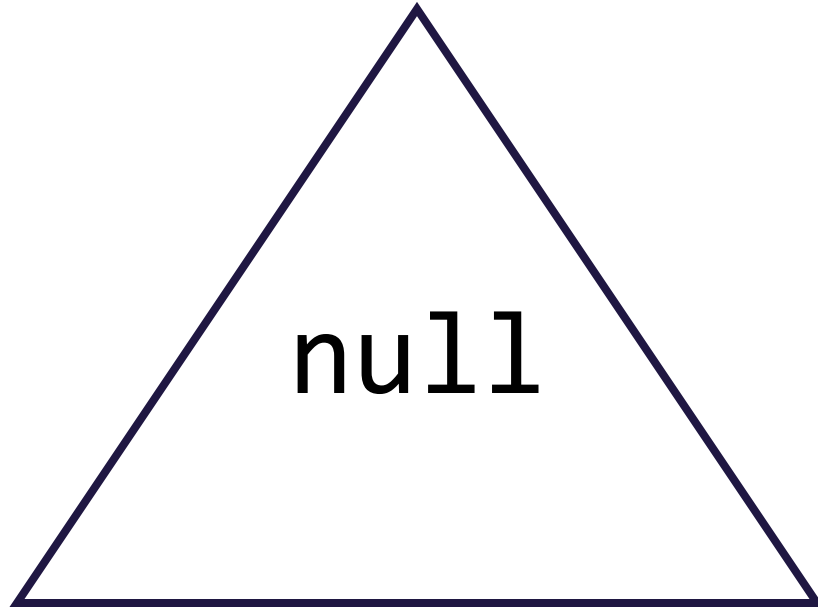


Announcements

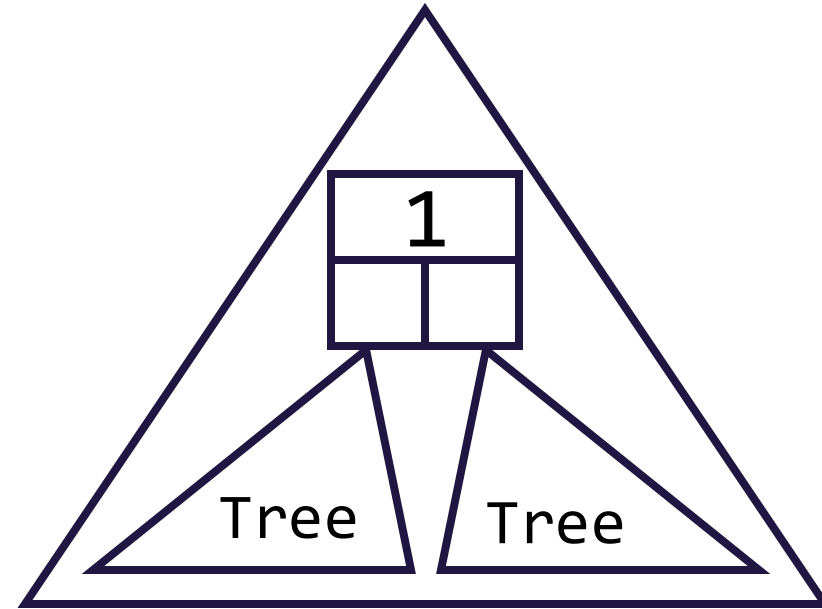
- Quiz 1, C2, and R5 feedback released!
- Check out your “receipt” on Canvas
- R6 due Friday
- P2 due tonight
- P3(!) is next will release after class on Friday (11/15)

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 $x = \text{change}(x)$
 - Don't stop early, return updated subtree (`IntTreeNode`)
- Let's trace through an example together...

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