

Traversal of Binary Trees

Preorder Traversal

PREORDER (T, p)

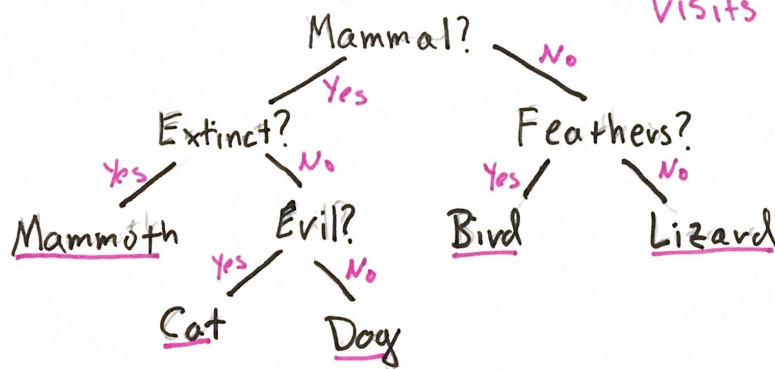
visit node at position p

PREORDER (T, p.left)

PREORDER (T, p.right)

T is a binary tree.

p is a position



Visits in this order:

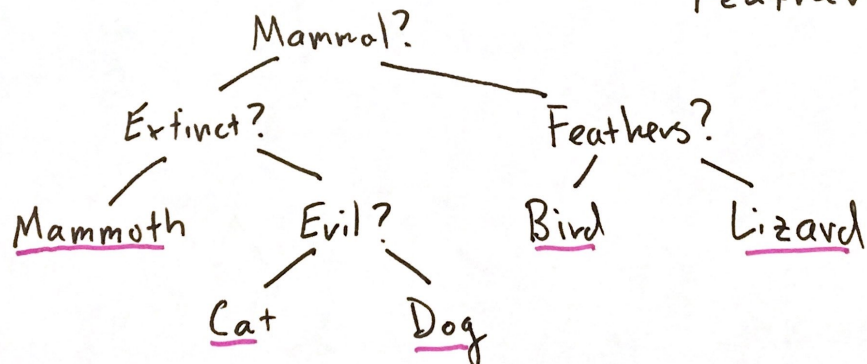
Mammal, Extinct,
Mammoth,
Evil, Cat, Dog,
Feathers,
Bird, Lizard

Postorder Traversal

POSTORDER (T, p)
POSTORDER (T, p.left)
POSTORDER (T, p.right)
visit node at position p

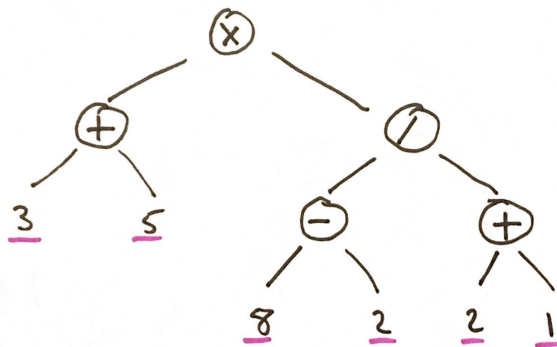
Visits in this order:

Mammoth, Cat, Dog,
Evil, Extinct,
Bird, Lizard,
Feathers, Mammal



Evaluation of Arithmetic Expressions

$$(3+5) \times ((8-2) / (2+1))$$



Use Postorder traversal.

Why?

What should "visit" do?

What is the tree for the following expression:

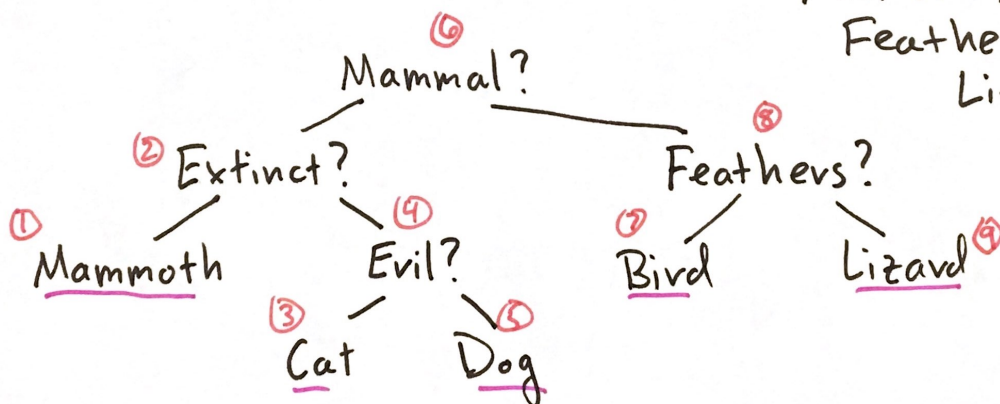
$$2^{5.7} \bmod 17$$

Inorder Traversal

INORDER (T, p)
INORDER (T, p.left)
visit node at position p
INORDER (T, p.right)

Visits in this order:

Mammoth, Extinct,
Cat, Evil, Dog,
Mammal, Bird,
Feathers,
Lizard



Traverses the tree from left to right.

Binary Search Trees

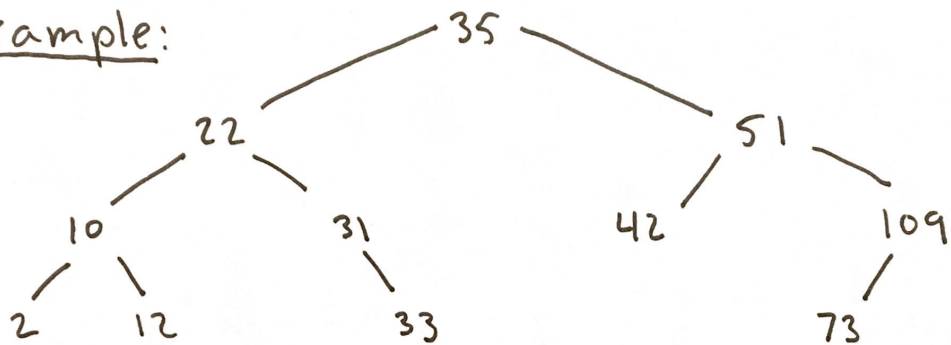
* If elements are ordered
(i.e. " $<$ " makes sense)

* Construct T so that for ^{any} position p :

(a) All elements of the subtree rooted at $p.left$ are smaller than the node at p .

(b) All elements of the subtree rooted at $p.right$ are larger than the node at p .

Example:

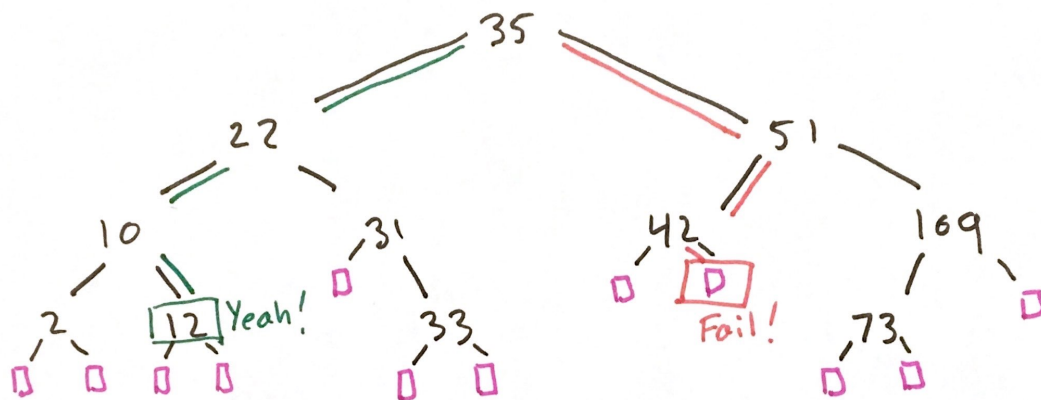


What does Inorder Traversal produce?
(Suppose "visit" just prints the value)

2, 10, 12 --- ??

Imagine that each leaf node actually has two "null" children.

Also, any internal node with only one child has a second "null" child.



The green path shows a successful search for the value 12.

The red path shows an unsuccessful search for the value 45.

What is the asymptotic (Big-Oh) running time of searching a BST?

Euler Tour Traversals

This is a **really cool generalization** of Preorder, Postorder, and Inorder.

EULERTour(T, p)
→ visit p "on the left"
if p is internal
 EULERTour($T, p.\text{left}$)
→ visit p "from below"
if p is internal
 EULERTour($T, p.\text{right}$)
→ visit p "on the right"

Three different
"visit" operations!

Leaf nodes do
all three visits
without any
recursive calls.

PREORDER is EULERTour with just "on the left" visits.

POSTORDER is...??

INORDER is...??

Euler Tour Traversals

This is a **really cool generalization** of Preorder, Postorder, and Inorder.

EULERTour(T, p)
→ visit p "on the left"
if p is internal
 EULERTour($T, p.\text{left}$)
→ visit p "from below"
if p is internal
 EULERTour($T, p.\text{right}$)
→ visit p "on the right"

Three different
"visit" operations!

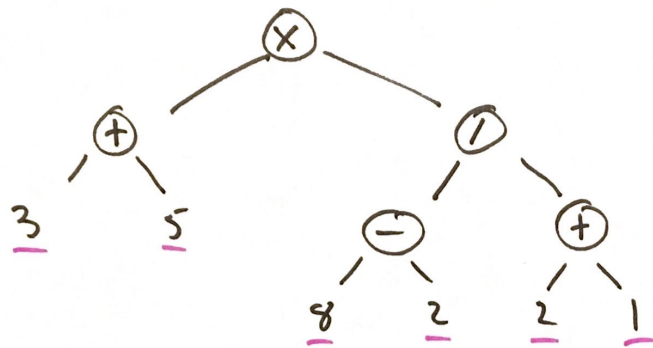
Leaf nodes do
all three visits
without any
recursive calls.

PREORDER is EULERTour with just "on the left" visits.

POSTORDER is...??

INORDER is...??

Printing Arithmetic Expressions



Perform Euler Tour Traversal with...

"On the left": if internal, print "("

"From below": print value or operator

"On the right": if internal, print ")"
