Lecture 20 - Wednesday, March 22

Announcements

- WrittenTest2 results to be released by FRI, March 24
- Assignment 3, ProgTest2
- Makeup Lecture for WrittenTest2
  + Expected to complete by: Exam Day

Lecture

Binary Trees ADT

Definition, Terminology, Properties

BT Properties: Relating #s of Ext. and Int. Nodes

n_E, n_I before ext.

n_E', n_I' after the ext.

Given a *binary tree* that is:

- ***nonempty*** and ***proper***
- with n_I ***internal nodes*** and n_E ***external nodes***

We can then expect that: $n_E = n_I + 1$ ✓

(# of external nodes; # internal nodes)

Induction on Size of Proper BT

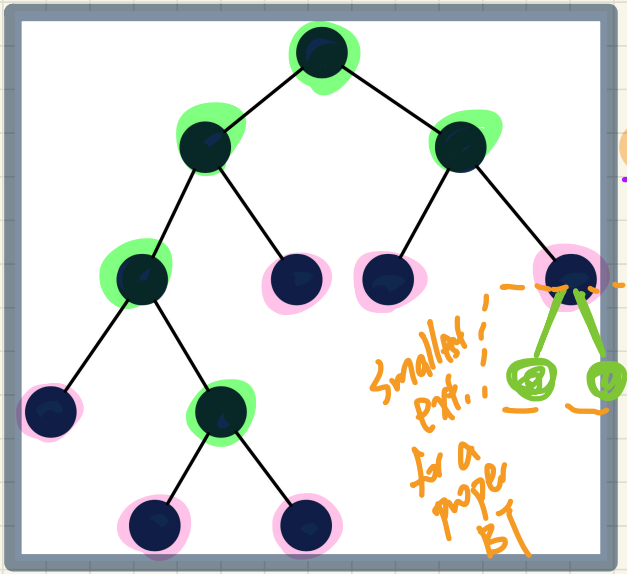
REVIEW

Base Case: 1-node tree

$n_E = 1$

$n_I = 0$

$n_E = n_I + 1$ ✓



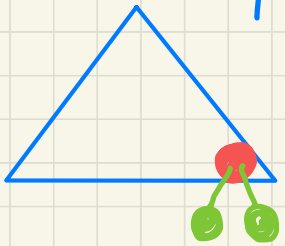
* By I.H.:

$n_E' = n_E + 1 = (n_I + 1) + 1 = n_I' + 1$

Inductive Hypothesis: for a proper BT with > 1 nodes: $n_E = n_I + 1$

Given a proper BT, extend it by turning an ext. node into an int. node $n_E' = n_E - 1 + 2 = n_E + 1$

$n_I' = n_I + 1$

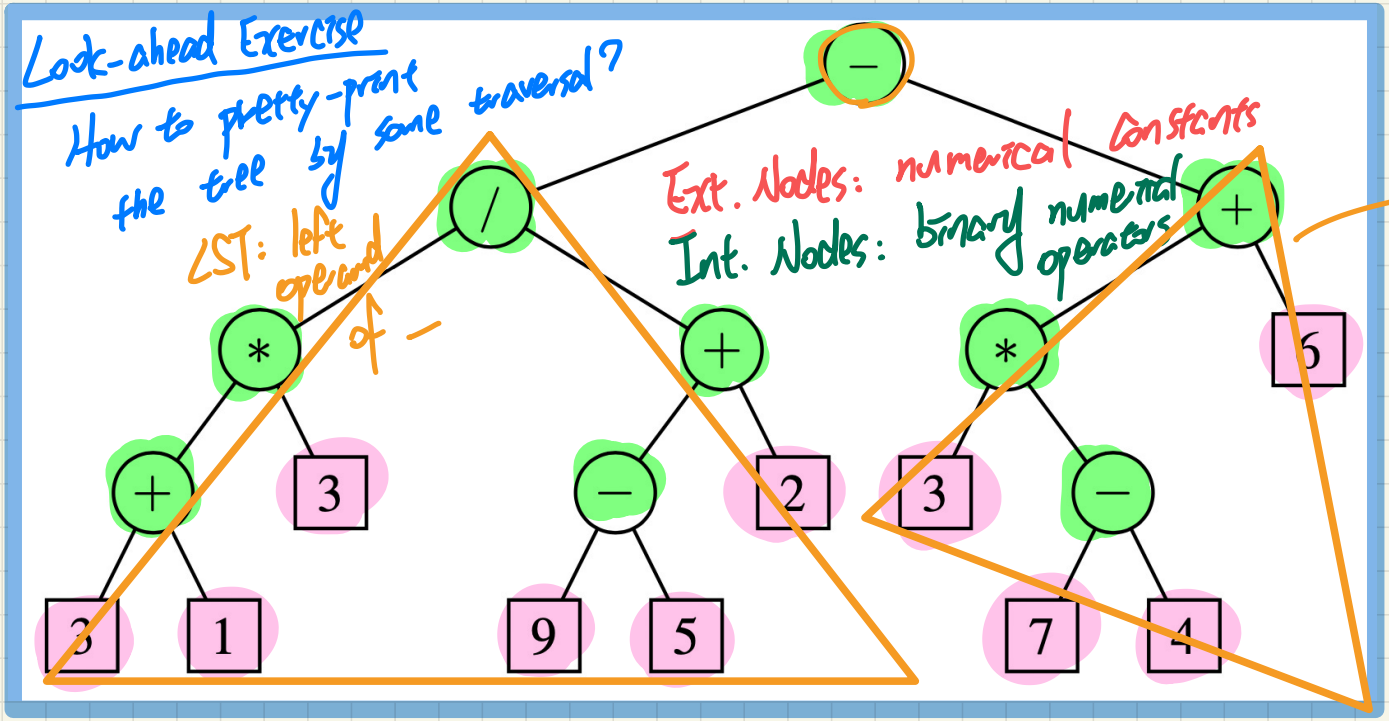


Lecture

Binary Trees ADT

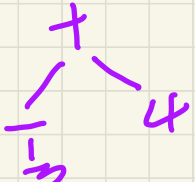
Applications

Applications of Binary Trees: Infix Notation



Q. Is the binary tree necessarily proper?

unary op. ← (−3) + 4

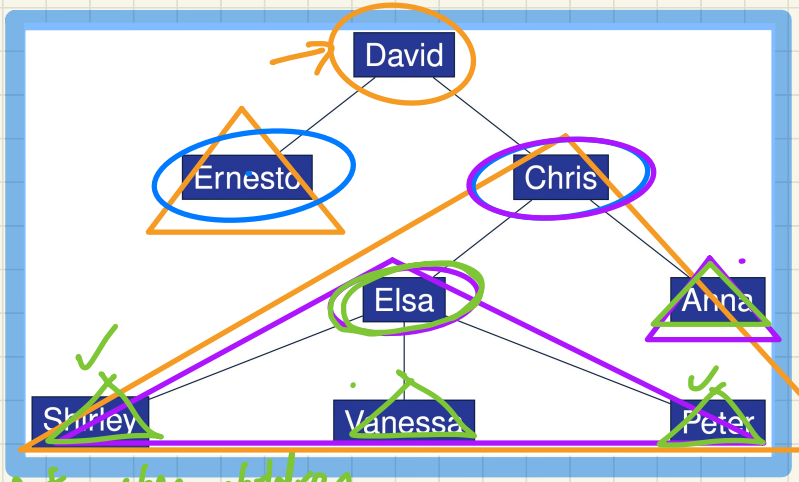


Lecture

Binary Trees ADT

Tree Traversals
Pre-Order, In-Order, Post-Order

General Tree Traversals: Pre-Order vs. Post-Order

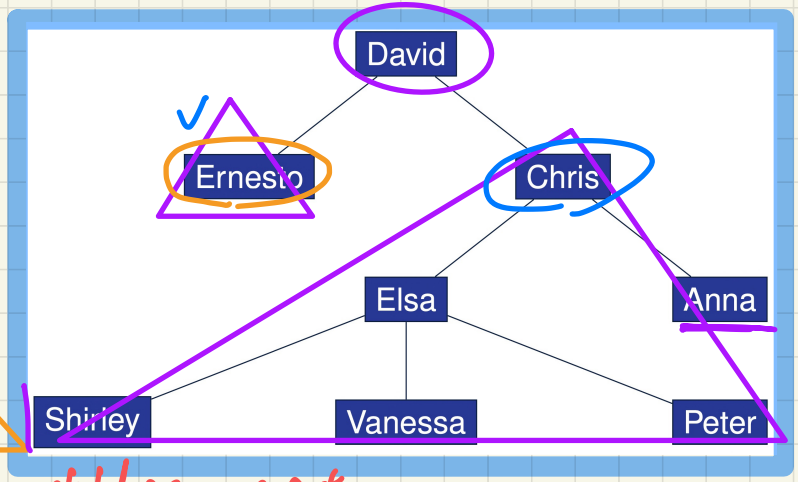


parent, then children

Pre-Order Traversal from the Root

pre-order (David)

David Ernesto → po(Ernesto) Chris Elsa Shirley Vanessa Peter → po(Elsa) Anna → po(Anna) → po(Chris)



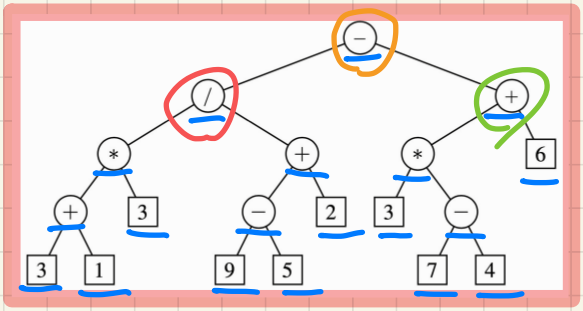
children, parent

Post-Order Traversal from the Root

post-order (David)

Ernesto → po(Ernesto) S. V. P. Elsa → po(Chris) A. C. David

Binary Tree Traversals

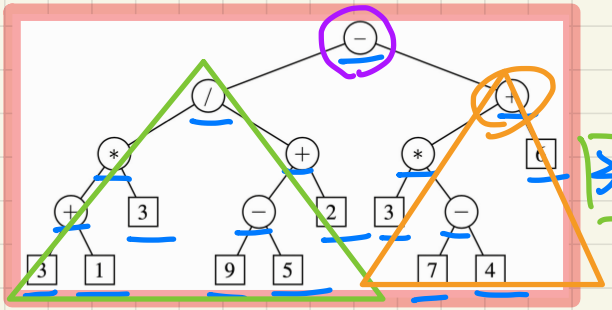


Pre-Order Traversal

− / * + 3 1 3 + − 9 5 2 + * 3 − 7 4 6

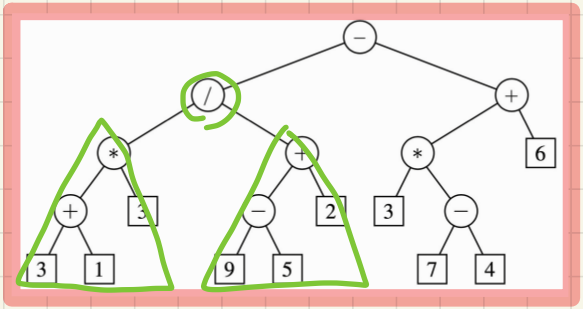
po(/)

po(+)



In-Order Traversal

3 + 1 * 3 / 9 − 5 + 2 − 3 * 7 − 4 + 6



Post-Order Traversal

postfix notation!

3 1 + 3 * 9 5 − 2 + / 3 7 4 − * 6 + −