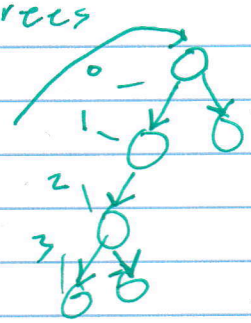
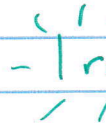
Chapter 18: Trees



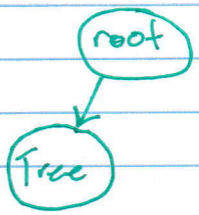
height of 3
depth 0



parent for children

leaves - no children

Recursively



Preorder - when you visit node, do first nodes then children

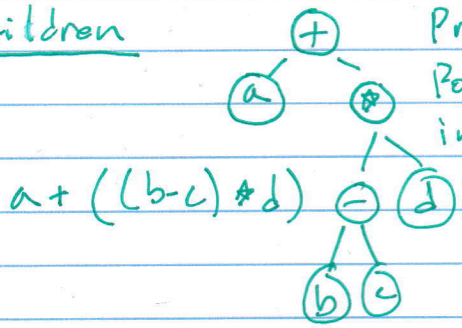
in order - left yourself right

Post order - for calculating things after children then node

with file sizes

level order

Binary Tree - 2 children
- left
- right



Pre - Print (- BC)

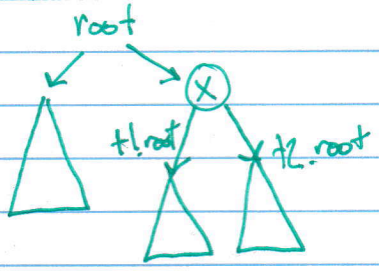
Post - Print (BC-)

in - Print (B-C)

Binary Node

left Binary Node right Binary Node

Merge



Aliasing:

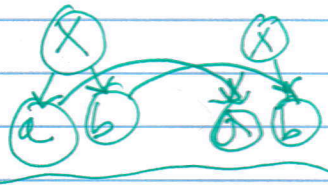
t1.merge(t1, t2)

Chapter 18

Duplicate

#1 duplicate the root

#2 duplicat left & right



Size

- size of left + size of right + 1 (root)

Height

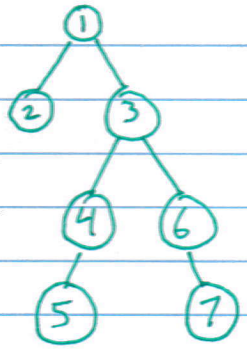
0 root

1 left right

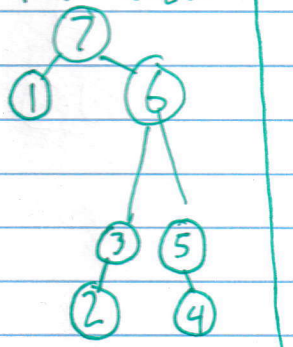
Max (left, right) +1

Tree Traversal

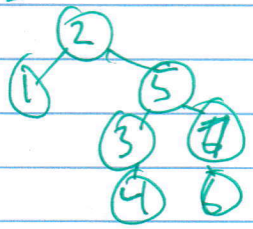
Pre-order



Post-order



In-order



Iterators

stacks can be used

Level order

root 1st

2nd

3rd

- Breadth first search