

Binary Search Trees

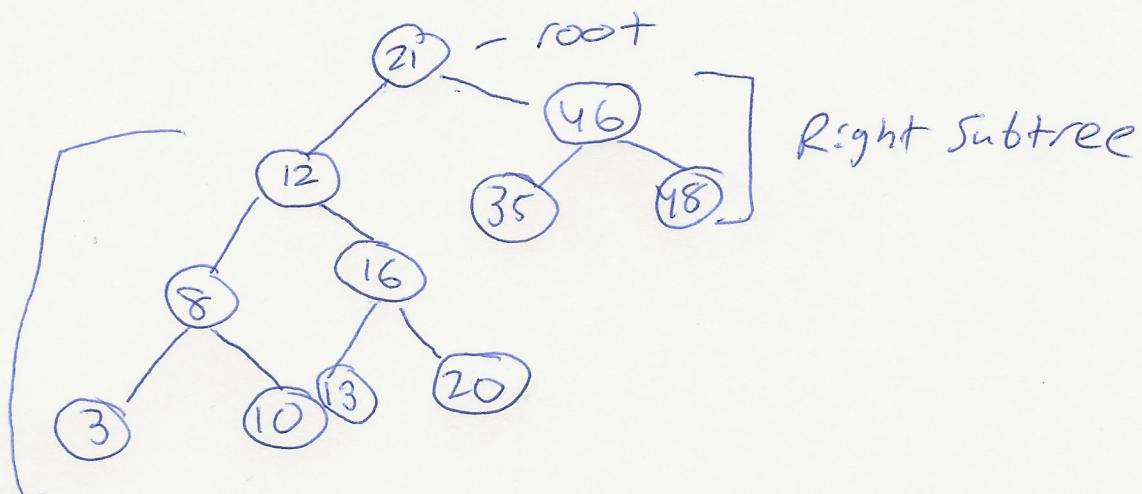
Asher Hamrick
CSE 17 Fall 2019

level 1

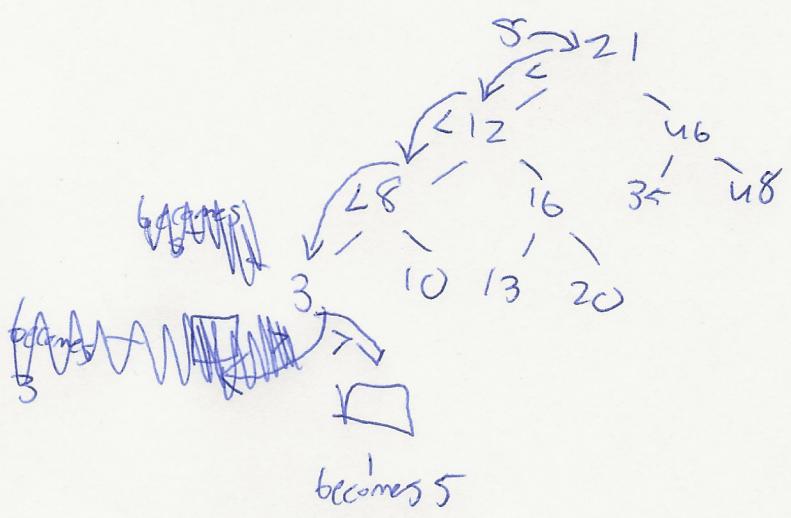
level 2

level 3

left subtree

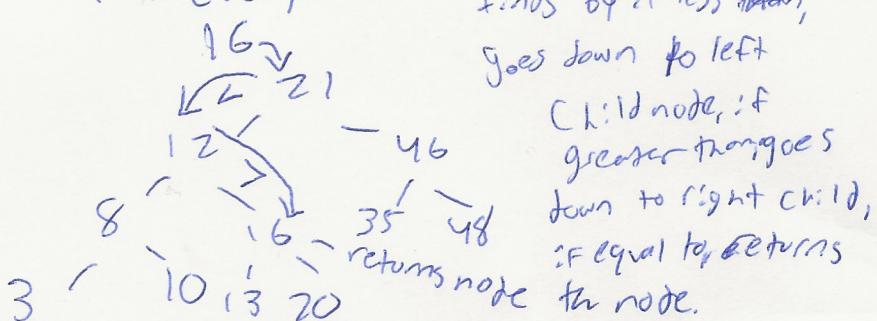


add(5)



compares added value to each node: if hits, if it is less than, moves to the left child, greater than, moves to the right child.

find(16)



finds by if less than,

goes down to left

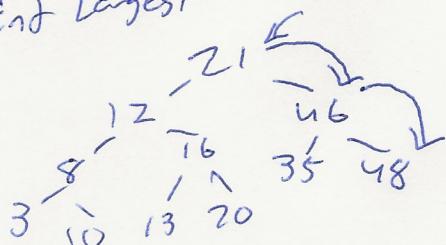
child node, if

greater than goes

down to right child,

if equal to returns

find largest(largest())



returns 48

All children on the right are greater than their parents, so by travelling down right every time, the rightmost and largest value is found.

delete(21)

To delete 21, its in-order traversal successor will become

new root. It is deleted and just replaces 21, with no children, a node can be removed, and without one child, it can be replaced by its child.

(contains) works the same way,

returning true if the node is found, false if it reaches a null child.



In-order

trav: 3, 8, 10, 12, 13, 16, 20, 21, 35