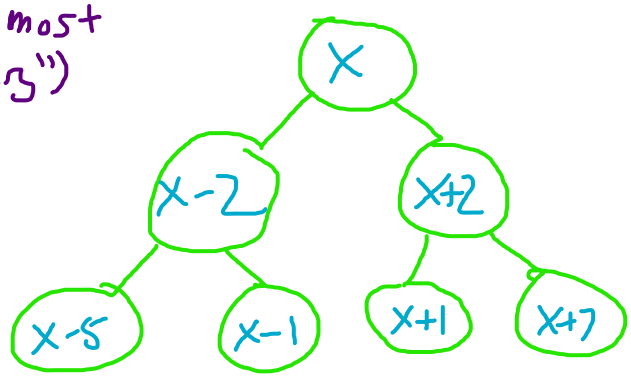
Binary Search Trees (BSTs)

BST Properties

- Every key in the left subtree of a node X is always less than X.
- Every key in the right subtree of a node X is always greater than X.
- Each node has at most 2 children ("binary")
- NO DUPLICATES

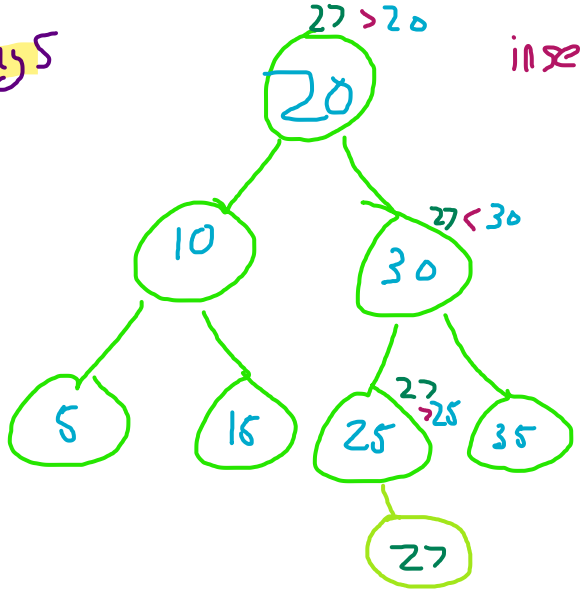


BST.insert(x)

1. Start at curr_node = root
2. if X > curr_node.value:
   curr_node = curr_node.right
   else:
   curr_node = curr_node.left

   } repeat until curr_node is NULL
3. insert new Node(X) at current location

★ New nodes always added as leaf



insert(27)

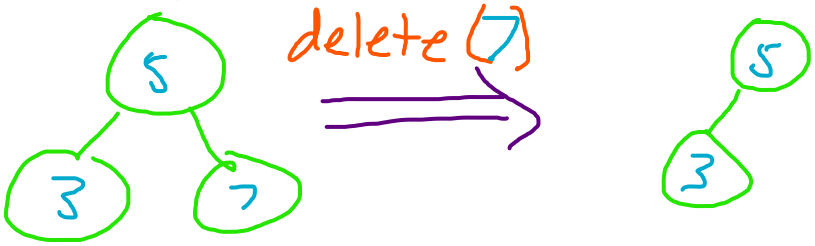
Challenge: how do we code this?

BST.delete(X)

3 cases:

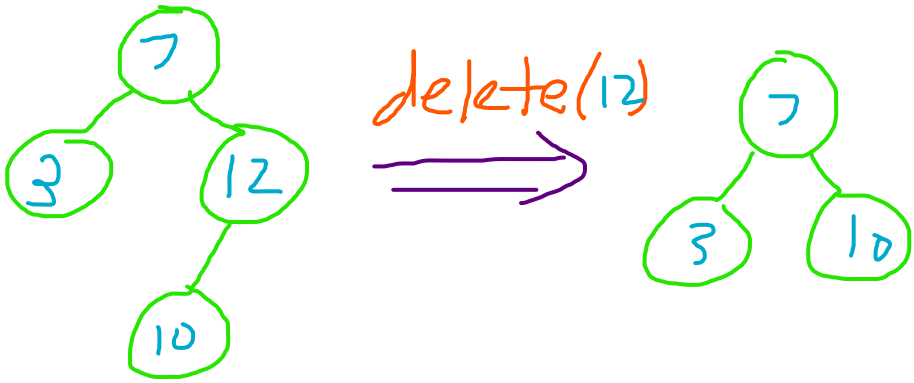
0 If node to be deleted has 0 children

just delete!



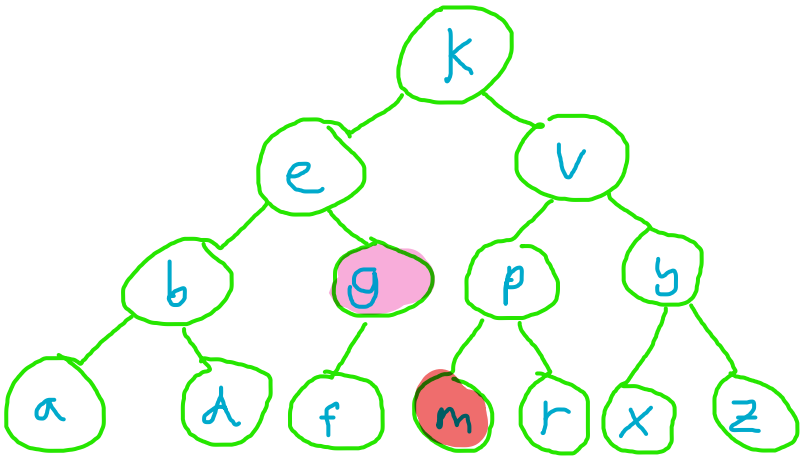
1 If node to be deleted has 1 child

link parent to child node to be deleted.



2 If node to be deleted has 2 children

to be deleted node is replaced by predecessor (largest node of left subtree of node) or successor (smallest node of right subtree of node)



delete(k)

or

delete(k)