

Recursion

- preorder traversal:
 - process current node
 - go left
 - go right

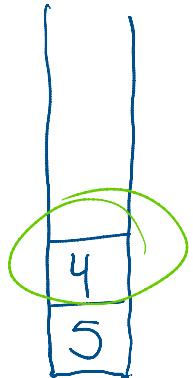
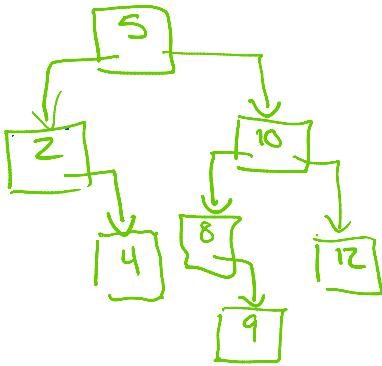
→ in-order traversal:

- * → go left
- process current
- go right

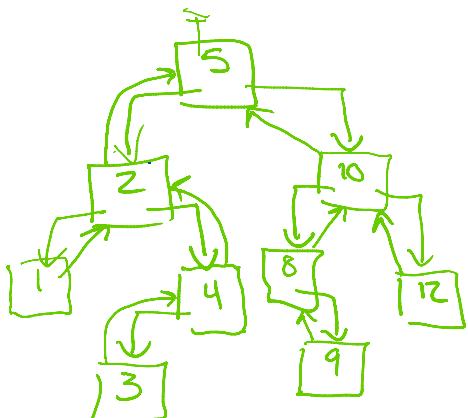
→ post order traversal:

- go left
- go right
- process current

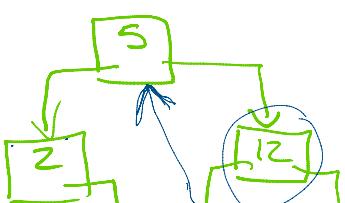
new iterator(3)

Stack

- if going left, push current node on stack
- if pop from stack, push new "next": leftmost of current node

Parent PointersFinding next

- if current has a right:
 - go to leftmost of right
- else:
 - if current is a left:
 - go to parent
 - else:
 - while current is a right:
 - go up
 - go up once more

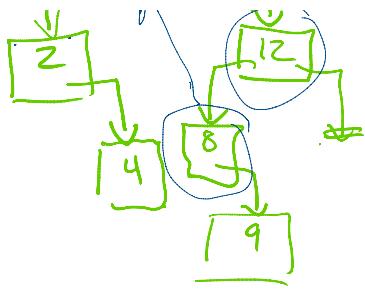
Remove

- node has no children:

→ parent's pointer to node
set to null

- node has one child:

→ parent's pointer to node



Pred: value before
Successor: value after

- node has one child.
 - parent's pointer to node set to that child
 - skip over removed node
- node has 2 children:
 - find either immediate predecessor or immediate successor
 - replace data in current with pred/suc
 - remove pred/suc