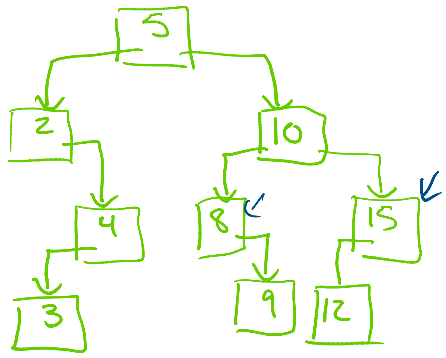
Week 9: BST Navigation and Removal

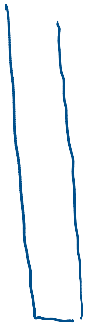
Friday, November 6, 2020 12:02 PM

Recursion

- pre-order:
  - process current
  - go left
  - go right
- *** in-order:
  - go left
  - process current
  - go right
- post-order:
  - go left
  - go right
  - process current

Stack

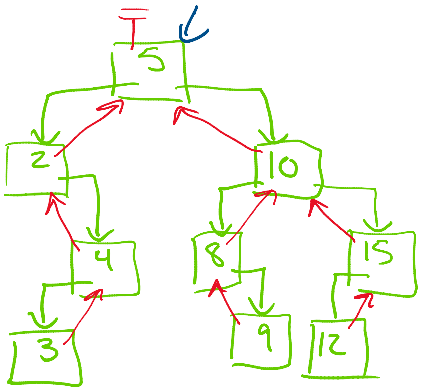




2,3,4,5,8,9,10,12,15 current: 15

- if going left, push current node on stack, then go left
- * if getting next current, pop off the stack.
  - push node after new current on the stack
    - go right while node is not null: push node, go left

Parent Pointers



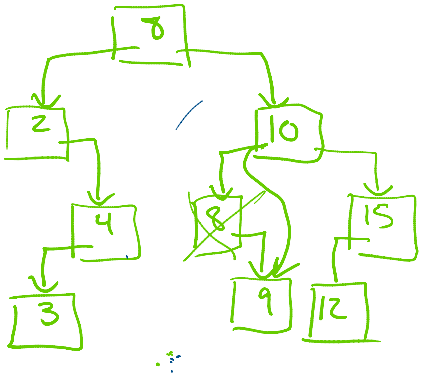
2,3,4,5

- * if current has a right child:
  - make current the leftmost of right
- * else:
  - * if current is a left child:
    - make current its parent
  - * else:
    - keep going up until no longer a right child
    - go up once more

Node class {
T data;
Node left, right, parent;
}

Remove

- if removing a node with no children:
  - set parent's left/right to null



-> if removing a node w/ no children:
  -> set parent's left/right to null

* -> if removing a node w/ 1 child:
  -> set parent's left/right to child

-> if node has 2 children:
  -> find either immediate predecessor (one before) or successor (one after)
  -> copy successor's data into node removing
  -> call remove on successor