* Delete Object from Binary Search Tree (BST)
  + Pseudo-code:
    - Node removeItem(Node root, Object toRemove) {
      * If (root is null)
        + return/do nothing
      * rootData = root.data;
      * if (toRemove == rootData)
        + root = removeFromRoot(root);

\*removeFromRoot(root) will delete the root node of the tree rooted at root argument.\*

* + - * Else if (toRemove is smaller than rootData)
        + Root.left = removeItem(root.left, toRemove);

Recursive call

* + - * Else
        + Root.right = removeItem(root.right, toRemove);

Recursive call

* + - * Return root; }
    - removeFromRoot(root) { //A more accurate name would be “removeRoot”
      * Case 1: The root has both left and right subtrees.
        + Step 1: Find the largest node in the left subtree
        + Step 2: Assign the largest Object data of the left subtree to be the new root
        + Step 3: Remove the old node from step 1
      * Case 2: If the root node only has a right subtree
        + Root = root.right
      * Case 3: If root node only has a left subtree
        + Root = root.left
      * \*Case 4: root has no children
        + Root = null
      * Return root; }