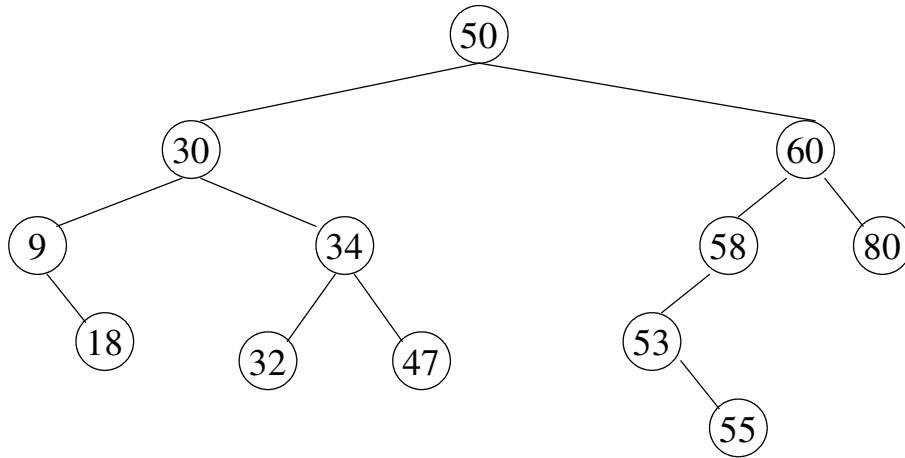


1. Consider the Binary Search Tree (BST):



- a. What would need to be done to delete 32 from the BST?
 - b. What would need to be done to delete 9 from the BST?
 - c. What would be the result of deleting 50 from the BST? Hint: One technique when programming is to convert a hard problem into a simpler problem. Deleting a BST node that contains two children is a hard problem. Since we know how to delete a BST node with none or one child, we can convert “deleting a node with two children” problem into a simpler problem by overwriting 50 with another node’s value. Which nodes can be used to overwrite 50 and still maintain the BST ordering?
 - d. Which node would the `TreeNode`’s `findSuccessor` method return for `succ` if we are deleting 50 from the BST?
2. When the `findSuccessor` method is called how many children does the `self` node have?
 3. How could we improve the `findSuccessor` method?
 4. When the `spliceOut` method is called from `remove` how many children could the `self` node have?
 5. How could we improve the `spliceOut` method?

6. The shape of a BST depends on the order in which values are added (and deleted).

a) What would be the shape of a BST if we start with an empty BST and insert the sequence of values:

70, 90, 80, 5, 30, 110, 95, 40, 100

b) If a BST contains n nodes and we start searching at the root, what would be the worst-case big-oh $O()$ notation for a successful search? (Draw the shape of the BST leading to the worst-case search)

7. We could store a BST in an array like we did for a binary heap, e.g. root at index 1, node at index i having left child at index $2 * i$, and right child at index $2 * i + 1$.

a) Draw the above BST (after inserting: 70, 90, 80, 5, 30, 110, 95, 40, 100) stored in an array (leave blank unused slots)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Index 0 Not Used																						

b) What would be the worst-case storage needed for a BST with n nodes?

8. a) If a BST contains n nodes, draw the shape of the BST leading to best, successful search in the worst case.

b) What is the worst-case big-oh $O()$ notation for a successful search in this “best” shape BST?