

Name: Marinna Ricketts-Uy

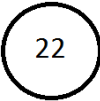
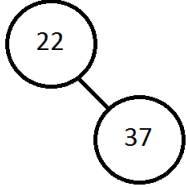
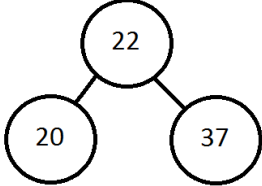
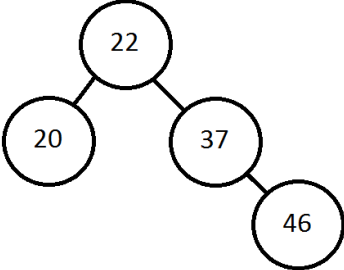
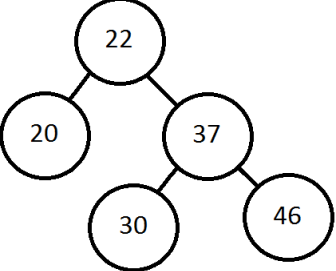
Section: 02 - Budhraj

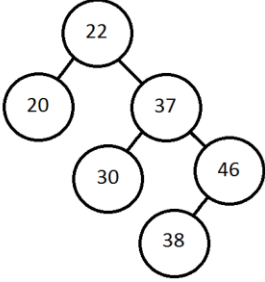
HW #: 3

Version: C

Username: pd12778

Insertion: Insert these numbers into the Binary Search Tree, 22 37 20 46 30 38.

Inserted	What the tree looks like	Explain what happened
22		The first item inserted will be the root of the Binary Search Tree.
37		Since the greater value goes to the right of a node, $37 > 22$ and is placed to the right of 22.
20		Since the lesser value goes to the left of a node, $20 < 22$ and is placed to the left of 22.
46		Since 46 is greater than 22, it will be on the right side of the root. 46 is greater than 37, so it is placed on the right side of 37.
30		Since 30 is greater than 22, it will be on the right side of the root. 30 is less than 37, so it is placed on the left side of 37.

38		<p>Since 38 is greater than 22, it will be on the right of the root. Next, 38 is greater than 37 so it will be on the right of the node 37. Finally, 38 is less than 46, so it will be placed on the left of the node 46.</p>
----	---	---

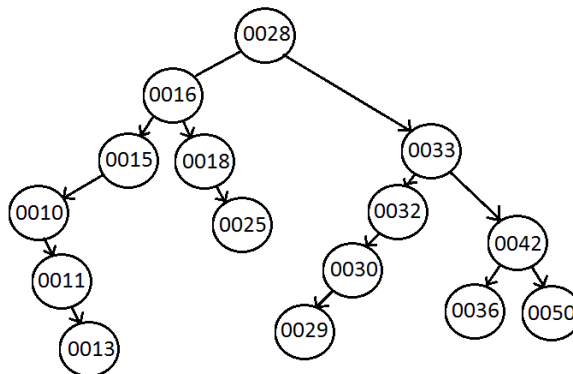
Searching:

Using the tree from above, the following are the steps the algorithm takes to search for the value 46:

1. Check if the value 46 is greater than or less than the root of the tree. In this case, the root is the value 22.
2. The value 46 is greater than 22 so 46 should be on the right subtree of the binary tree.
3. Next, look at the next value and see if it is equal to, less than, or greater than 46. The value is a 37, so 46 is greater than this value and would belong to the right of 37.
4. Then once again, check if the current value is equal to, less than or greater than 46. The current value is 46, so the value 46 has been found.

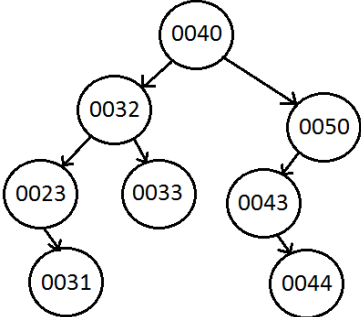
Deletion:

Original tree:



# to delete	Draw final tree after deletion	Why does it look this way?
28		25 is the closest number to 28 on the left subtree. Since 25 is a leaf, it can be moved to where 28 without having to rearrange the left subtree. Another option could have been 29 from the right subtree, since 29 is the smallest value in the right subtree and closest to 28.
11		Since 11 had a right child node, 11 will be deleted and replaced with the child node of value 13. So now 10 will have a right child node of the value 13.
50		Since 50 is just a leaf and doesn't have any children, it can just be deleted without anything else changing or being rearranged.

Pre/In/Post:

		
Pre-Fix Order	In-Fix Order	Post-Fix Order
40, 32, 23, 31, 33, 50, 43, 44	23, 31, 32, 33, 40, 43, 44, 50	31, 23, 33, 32, 44, 43, 50, 40