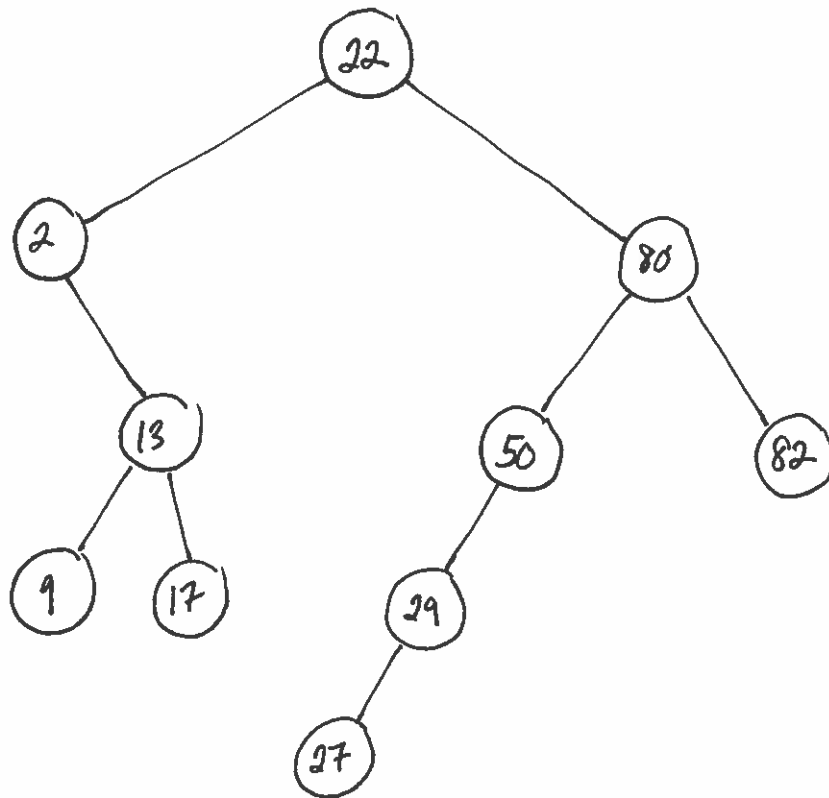


pre-order traversal

↑
root is processed before
left and right subtrees.

1. process the root of this (sub) tree
2. traverse left subtree with pre-order traversal
3. traverse right subtree with pre-order traversal



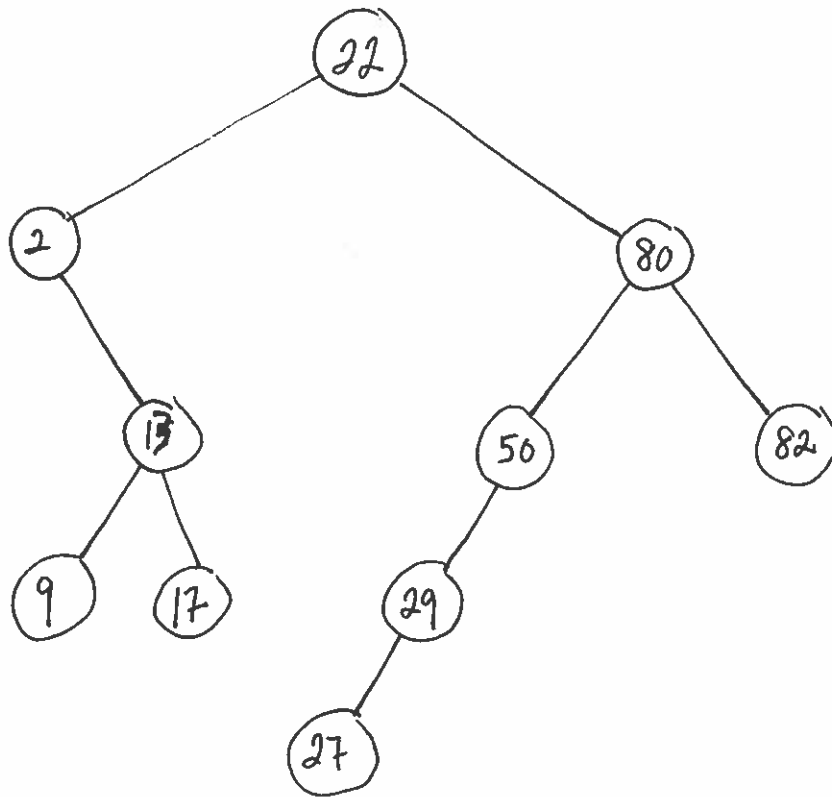
pre-order traversal: 22, 2, 13, 9, 17, 80, 50, 29, 27, 82

↖ * In a pre-order traversal, the root is always the first node to be processed.

post-order traversal:

↑
root is processed after
left and right subtrees

1. traverse left subtree with post-order traversal
2. traverse right subtree with post-order traversal
3. process the root of this (sub)tree



post-order traversal: 9, 17, 13, 2, 27, 29, 50, 82, 80, 22

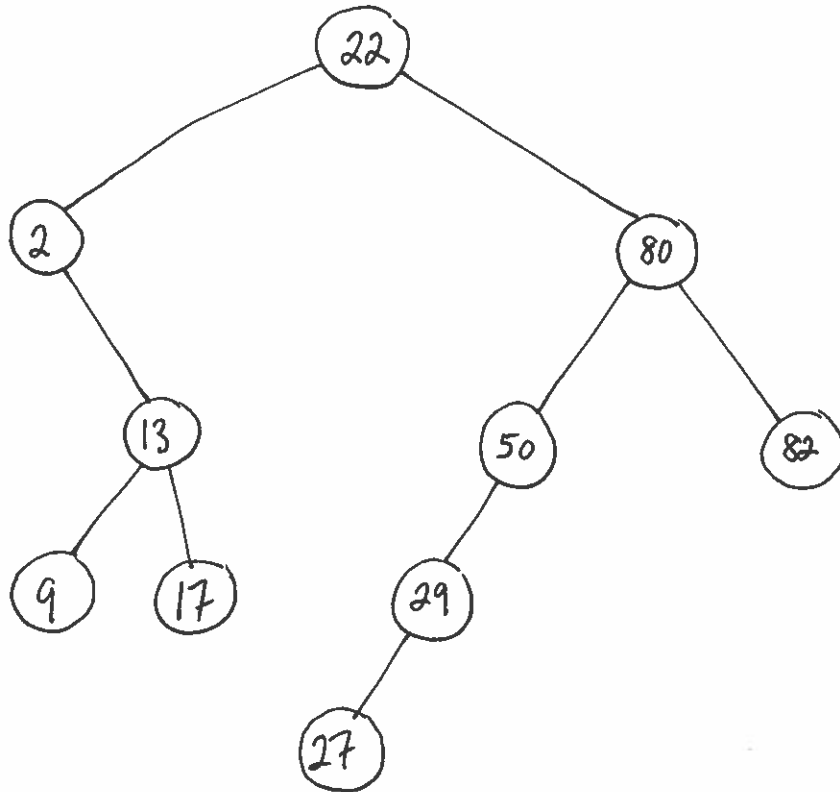


* In a post-order traversal, the root is always the last node to be processed.

in-order traversal:

↑
root is processed in between
left and right subtrees.

1. traverse the left subtree with in-order traversal
2. process the root of this (sub)tree
3. traverse the right subtree with in-order traversal



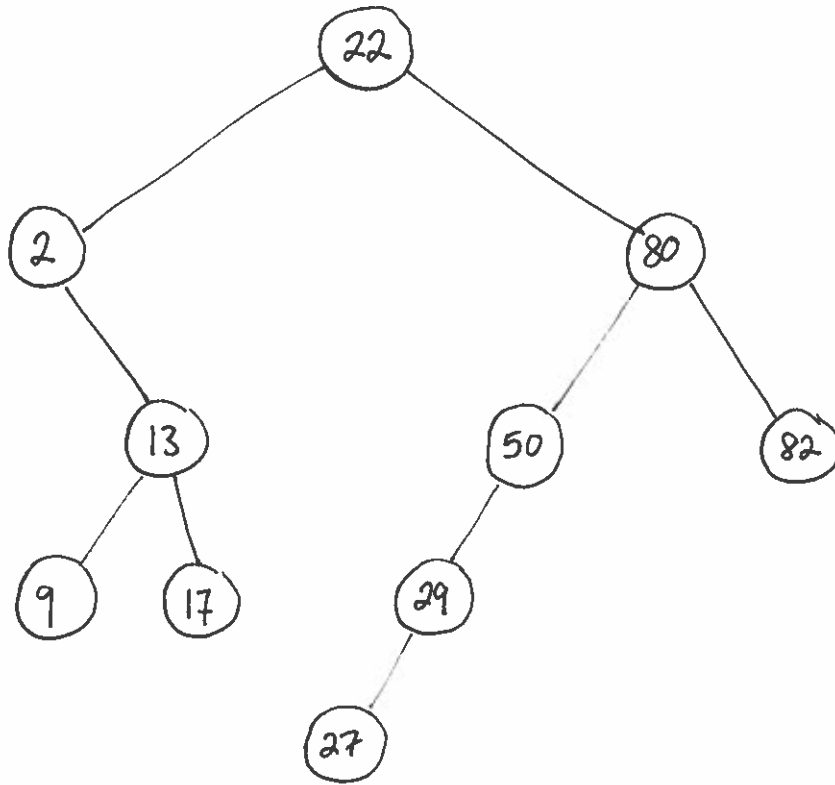
in-order traversal: 2, 9, 13, 17, 22, 27, 29, 50, 80, 82

↑
* Performing an in-order traversal on a binary search tree (not just an arbitrary binary tree) processed the values in our BST in sorted order.

level-order traversal:

↑
we go level by level

Go one level at a time, processing each node from left to right before moving on to the next level.



level-order traversal: 22, 2, 80, 13, 50, 82, 9, 17, 29, 27