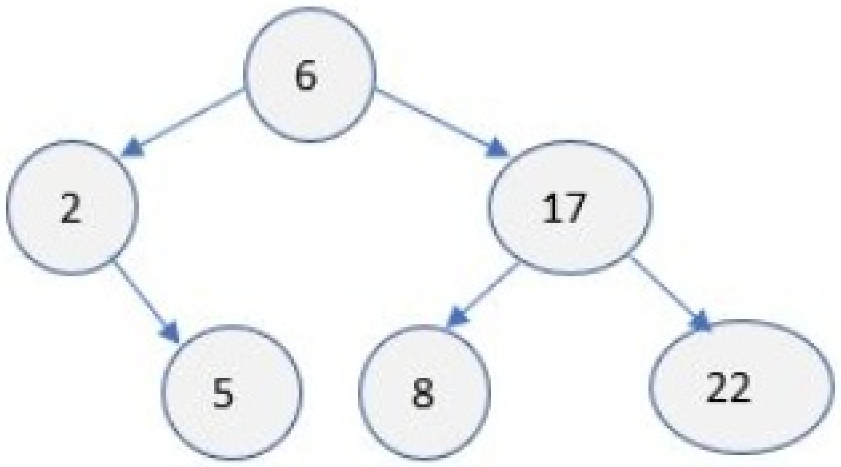
1. Suppose we wish to insert nodes with keys $-5, 11$ and 10, in that order, into the following Binary Search Tree (BST). **1 / 1 point**



Select all true statements from the list below.

☑ The node with key -5 will become a left child of the node with key 2.

✓ **Correct**

□ When the node with key 11 is inserted, it becomes the left child of the node with key 8.

□ When the node with key 11 is inserted, it displaces the node with key 8, which becomes its left child.

☑ When the node with key 10 is inserted, it becomes the left child of the node with key 11.

✓ **Correct**

1 / 1 point

2. Starting from an empty tree, we insert the nodes with keys $[1, \ldots, n]$ in some order. Select all the true statements from the list below.

☑ If the nodes are inserted in descending order, then the resulting tree has height n.

✓ **Correct**

☑ For $n = 7$, inserting the nodes in the order $[4, 2, 1, 3, 6, 5, 7]$ yields a fully balanced binary tree of depth 3.

✓ **Correct**

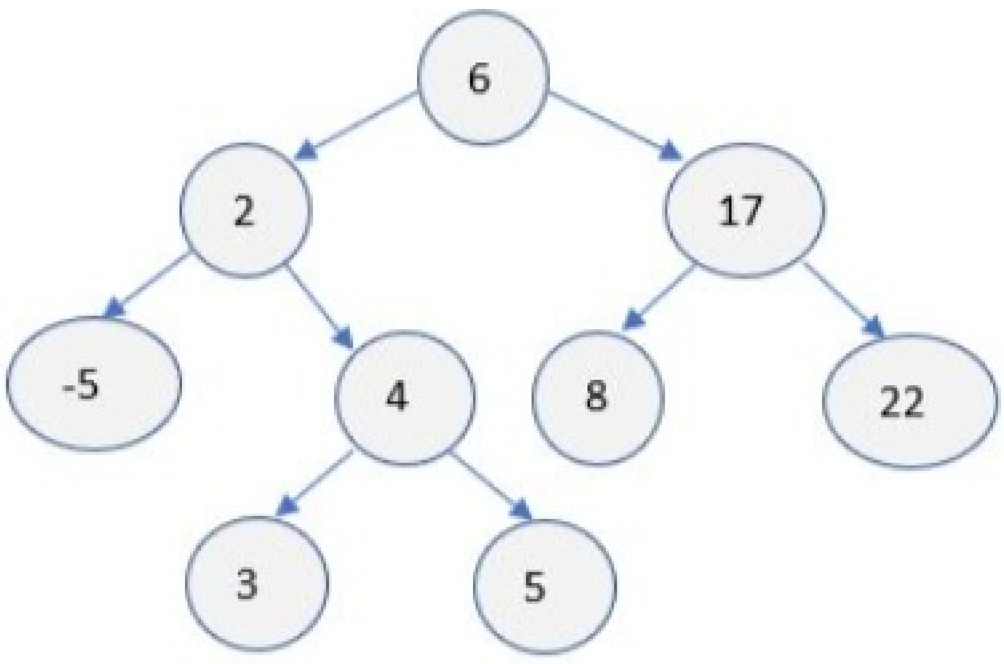
□ For $n = 7$, the only two insertion sequences that yield a tree of depth n are when the keys are inserted in ascending or in descending order.

☑ The tree can have depth between $\log n$ and n, depending on the actual order which the keys are inserted.

✓ **Correct**

3. Consider the following Binary Search Tree.

1 / 1 point



Select the single true statement from the list below.

○ If we delete the root (6), it will be replaced by one of its children.

○ If we wish to delete the root (6), its successor can be found by traversing its *leftmost* branch.

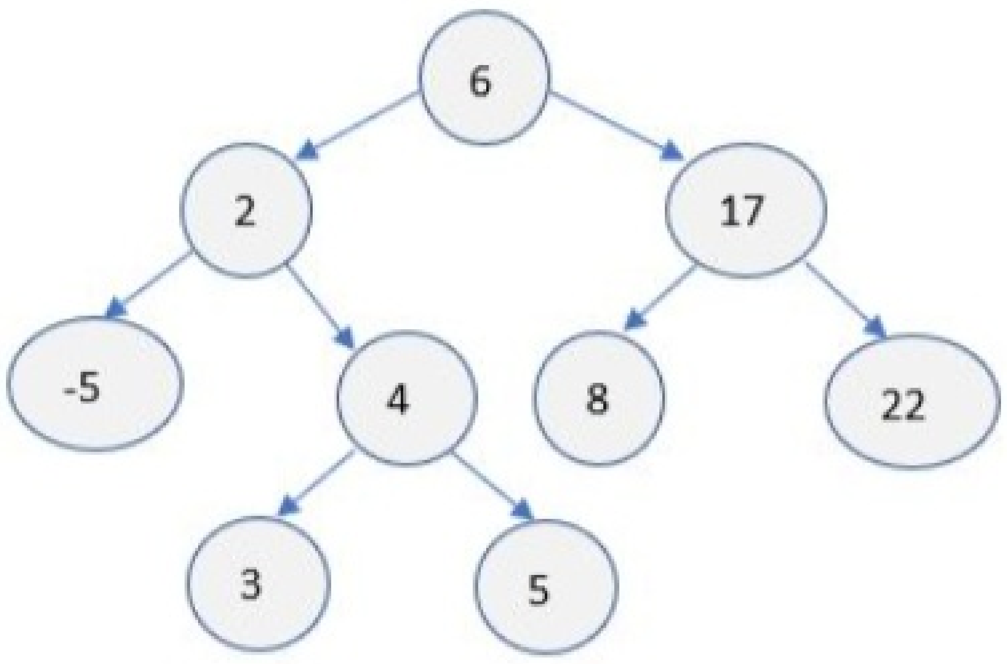
○ It is not possible to delete the root node from a tree since that will leave two disconnected subtrees.

● If we wish to delete the node 2, we can replace it with its successor node 3. In this case, the node 4 will be left with just one child.

Correct

4. Consider the Binary Search Tree below.

1 / 1 point



Select all the true statements from the list below.

☑ In-order traversal of a Binary Search Tree always leads to a sorted list of keys.

Correct
Correct – this is guaranteed by the binary search tree property.

☑ Pre-order traversal of the BST above yields the list $[6, 2, -5, 4, 3, 5, 17, 8, 22]$.

Correct

□ Post-order traversal of a BST produces the reversal of the list obtained from its pre-order traversal.

☑ All traversals require as much time as the number of nodes in the tree.

Correct