



# Binary Search Tree Insertion

In this lesson, we'll study the binary search tree insertion algorithm!

We'll cover the following



- Binary Search Tree Insertion Algorithm
  - Explanation

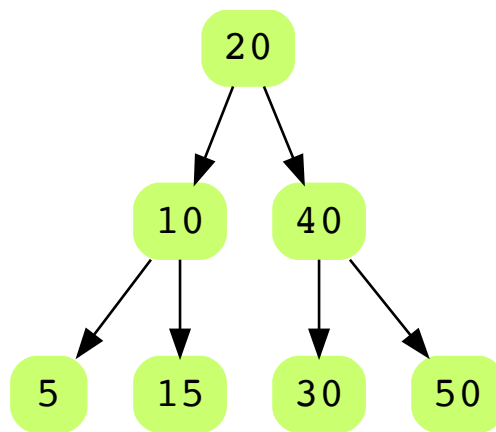
## Binary Search Tree Insertion Algorithm#

Here is a description of the algorithm you'd use to insert a new value into a BST.

1. Start from the root node
2. Check if the value to be inserted is greater than the root/current node's value
3. If yes, then repeat the steps above for the right subtree, otherwise repeat the steps above for the left sub-tree of the current node.
4. Repeat until you find a node that has no right/left child to move onto. Insert the given value there and update the parent node accordingly.

Study the animation below for a visual of this algorithm.





**Insert 17!**

1 of 11



## Explanation#

We first insert **17** into the tree. Starting from the root, we compare **17** with the value at the node, which is **20** and since **17** is smaller than **20**, we move on to the left sub-tree. Since the value there, **10**, is smaller than **17** we move on to the right sub-tree. The value here, **15** is also smaller than **17** so we move on to the right sub-tree. However, **15** has no right-child, so we simply create a new node with the value **17** and make that the right child of **15**. We will insert **12** by performing the same steps, starting from the root node.

In the next lesson, we will dive into the Python implementation of the BST insert algorithm which we studied in theory in this lesson. We will show you two flavors of the algorithm: iterative and recursive.



Interviewing soon? We've partnered with Hired so that

companies apply to you instead of you applying to them.

[how](#) ⓘ



← Back

Next →

Implementing a Binary Search Tree in ...

Binary Search Tree Insertion (Impleme...

✓ Completed

⚠ Report an Issue

