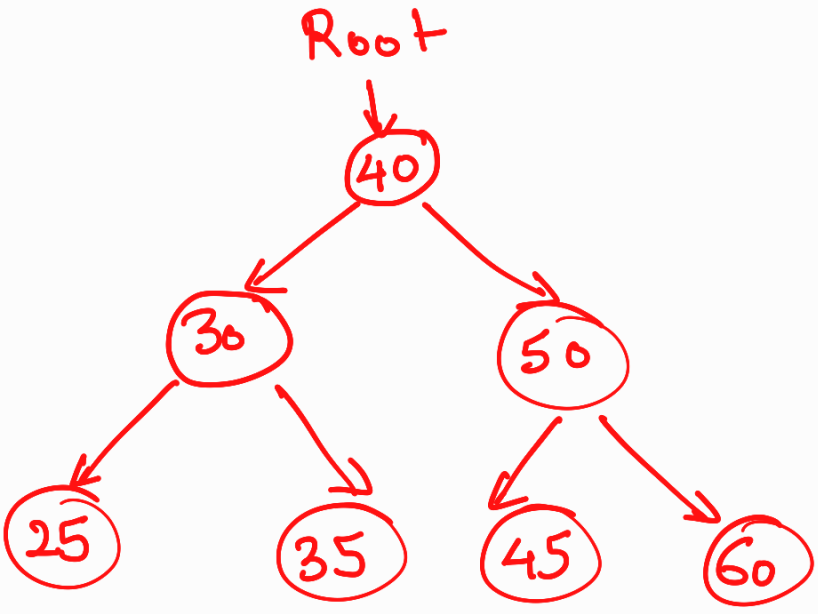
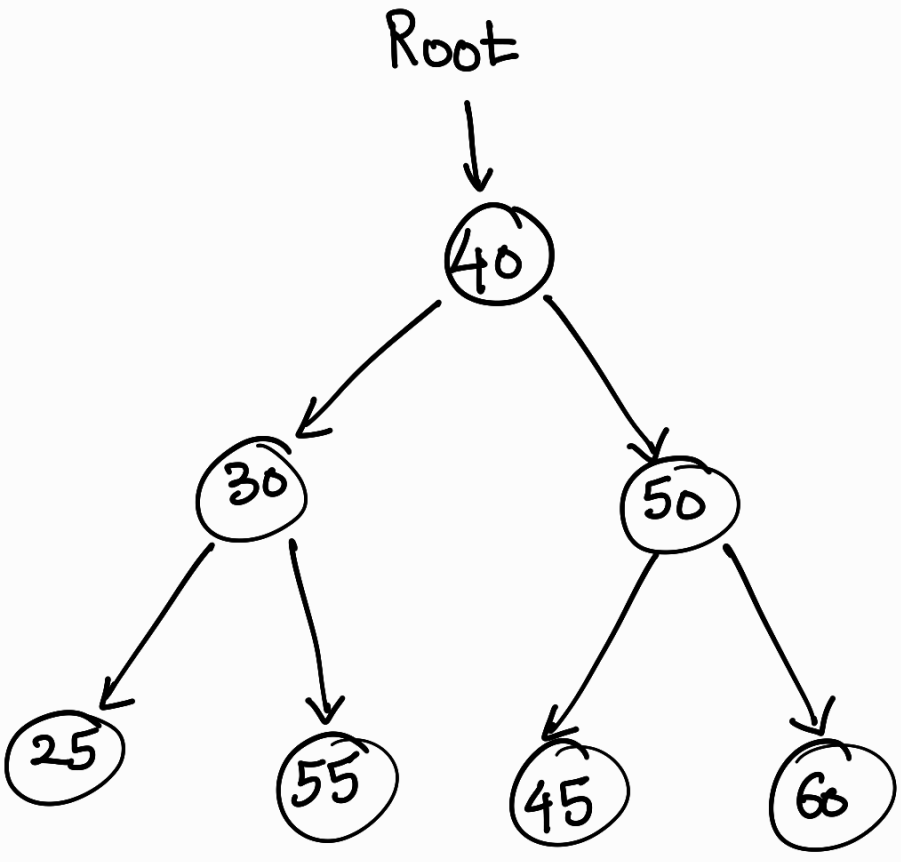
Binary Search Tree

A Binary search tree follows some order to arrange the elements.

In a Binary search tree, the value of left node must be smaller than the parent node, and the value of right node must be greater than the parent node.

This rule is applied recursively to the left & right subtree of the root.





Is it a Binary Search Tree ??

Advantages :

* Searching an element in the Binary search tree is easy as we always have a hint that which subtree has the desired element
* As compared to array & linked lists, insertion & deletion operations are faster.

Example

Suppose the data elements are :-

45, 15, 79, 90, 10, 55, 12, 20, 50

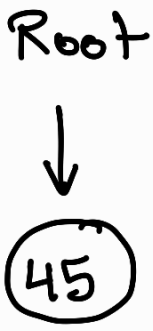
Sol:

1. First we have to insert 45 into the tree as the root of the tree.

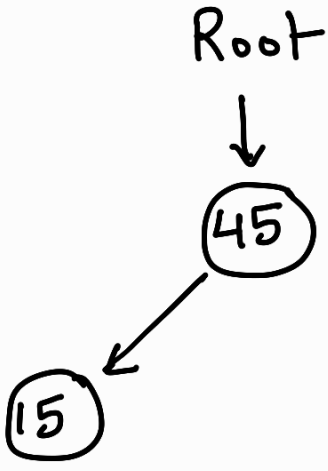
2. Then, read the next element; if its smaller than the root node, then insert it as the root of the left subtree, and move to the next element.

3. Otherwise, if the element is larger than the root node, then insert it as the root of the right subtree.

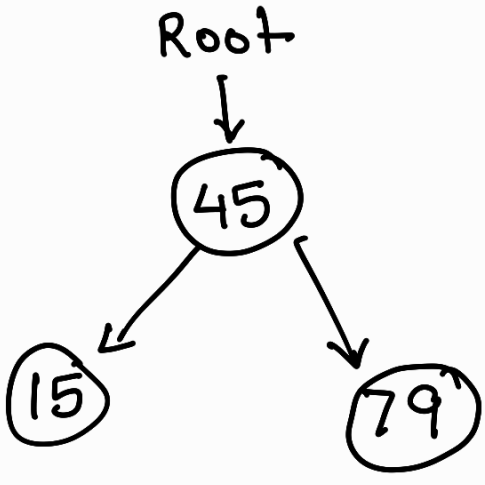
Step 1 . Insert 45.



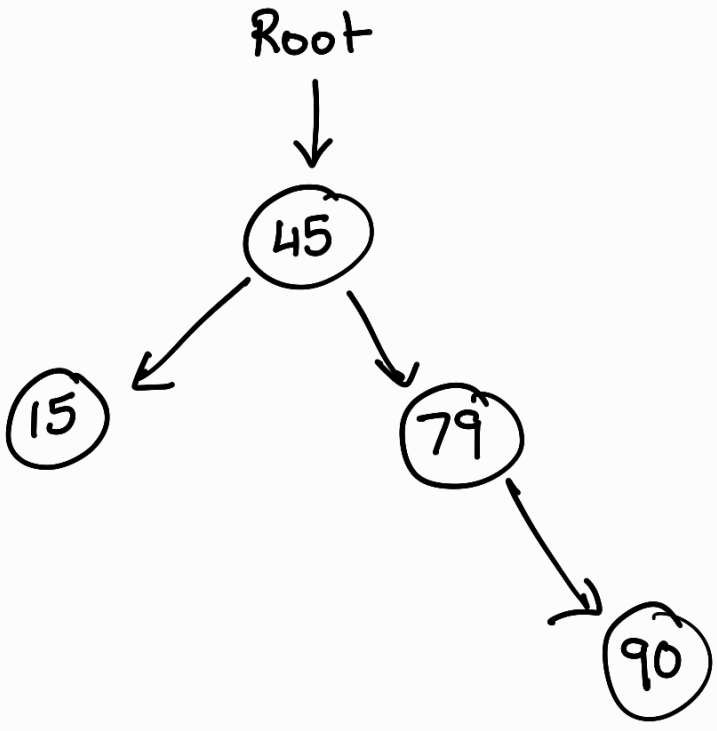
Step 2 Insert 15



Step3 Insert 79

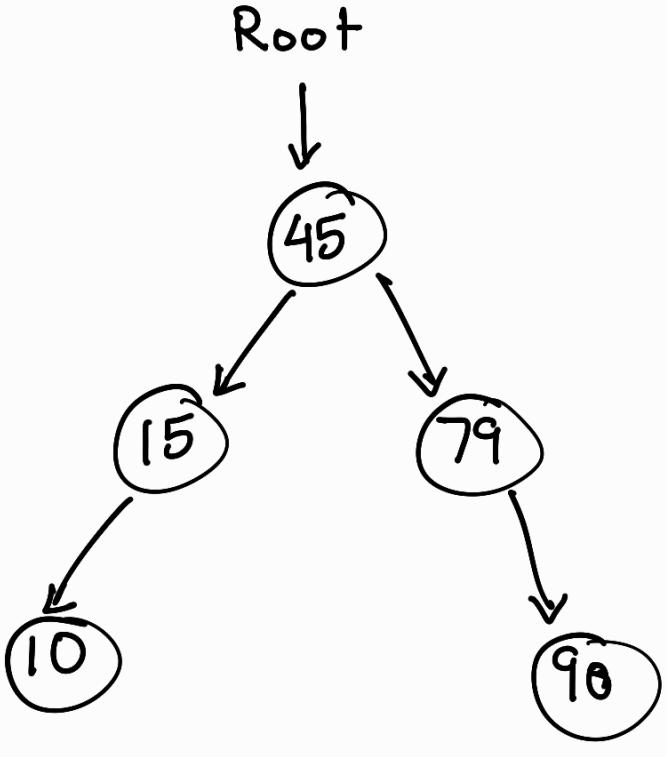


Step 4



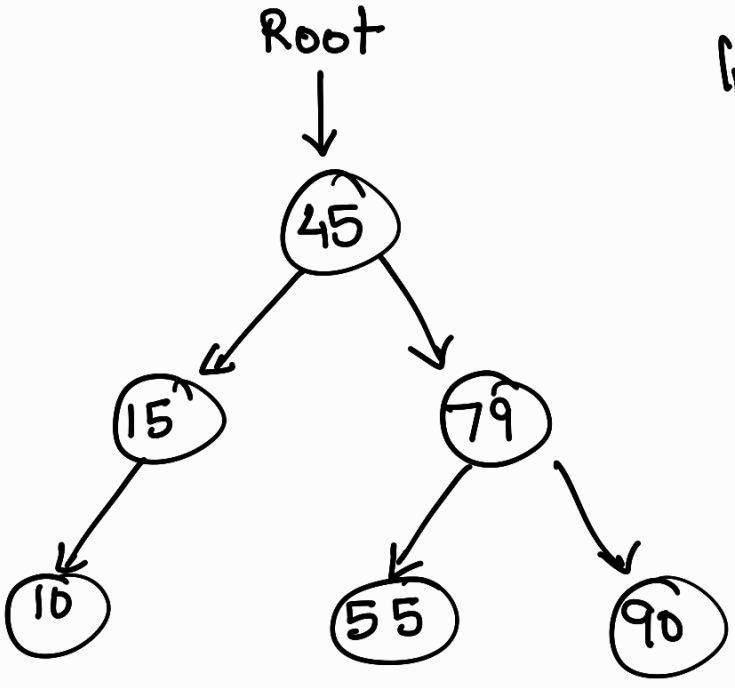
Insert 90

Step 5



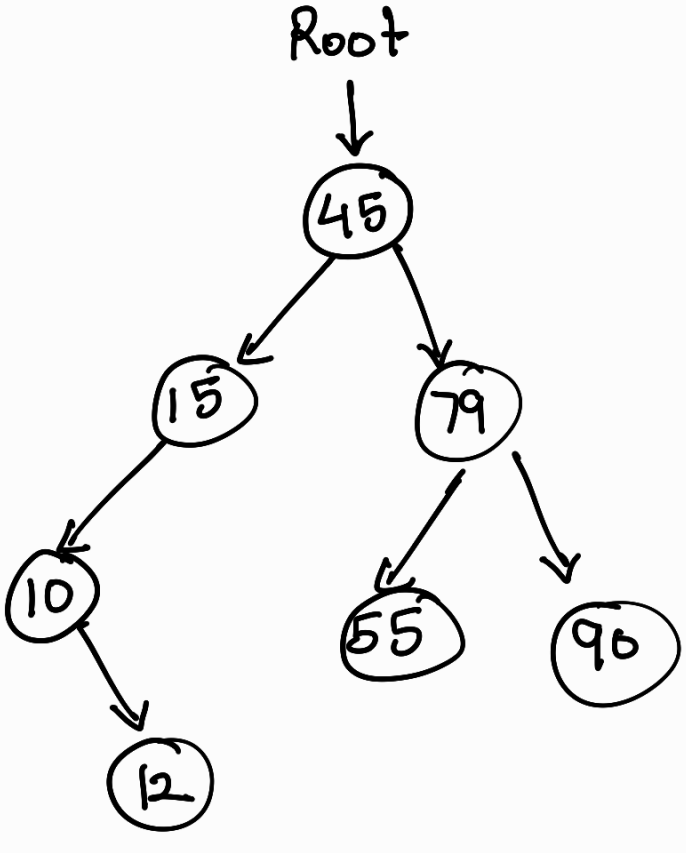
Insert 10

Step 6



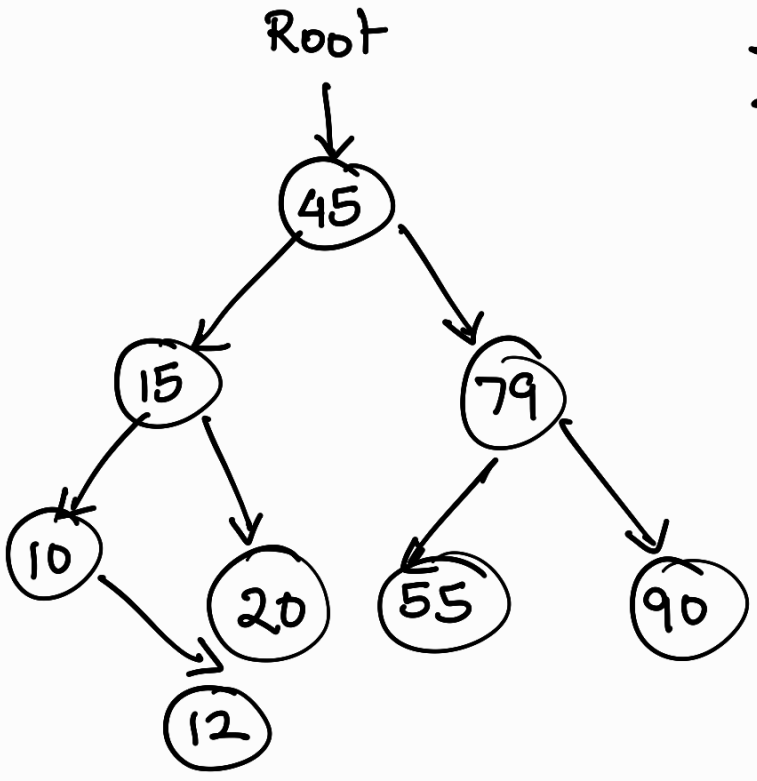
Insert 55

Step 7



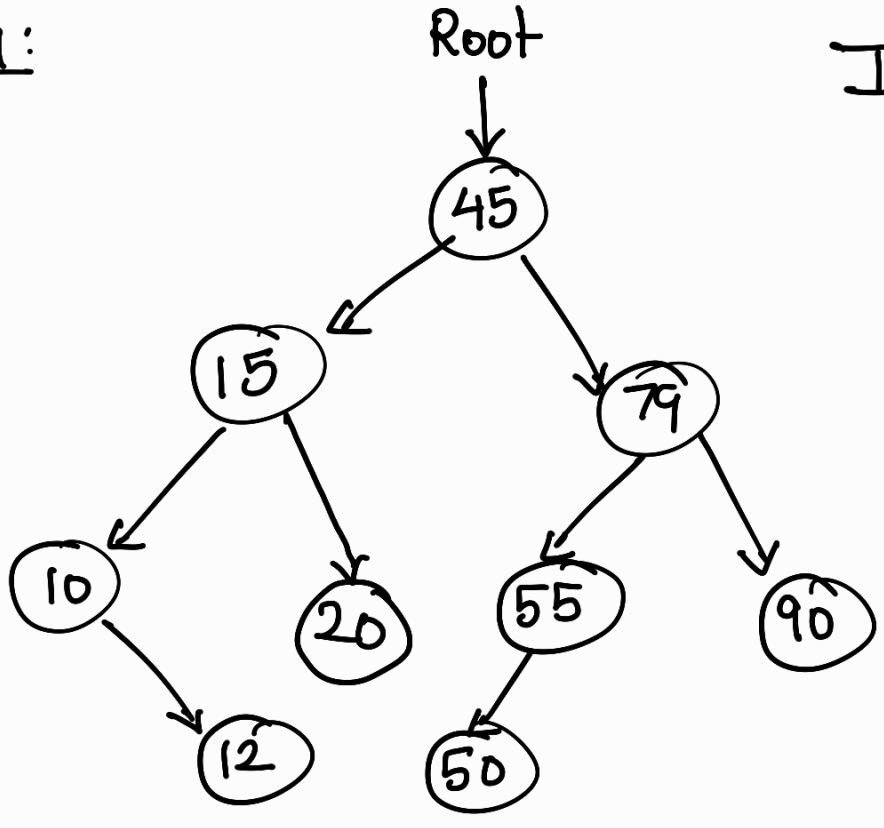
Insert 12

Step 8:



Insert 20

Step 9:



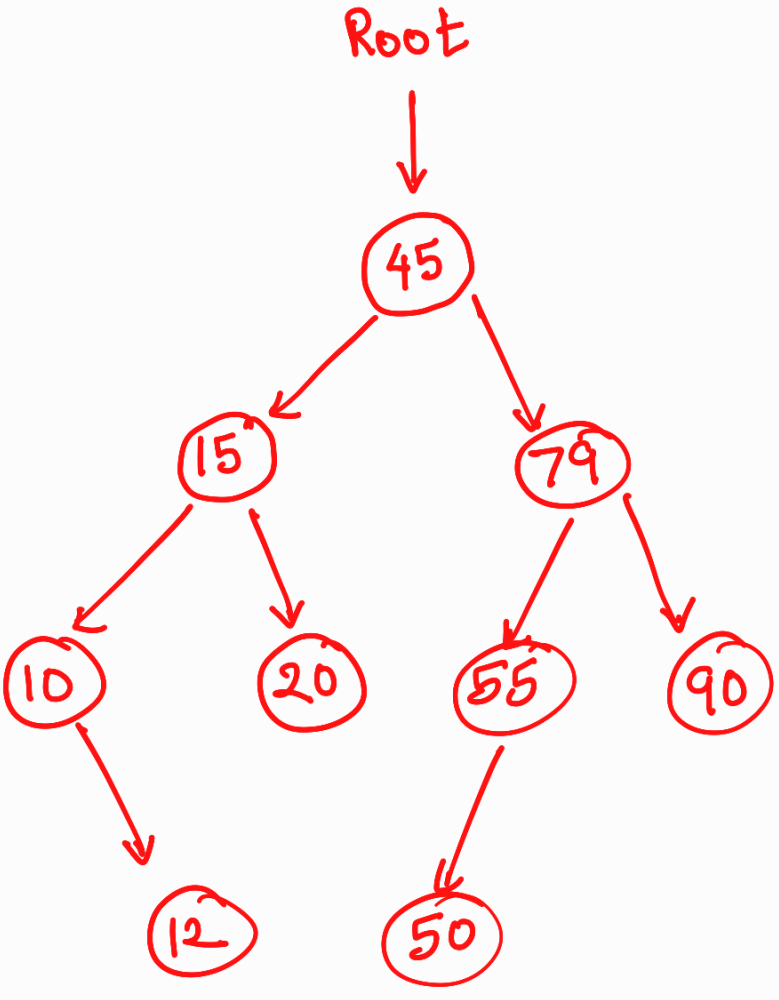
Insert 50

Searching in a Binary Search Tree

1. First, compare the element to be searched with the root element of the tree.
2. If root is matched with the target element, then return the node's location.
3. If it is not matched, then check whether the item is less than the root element, if it is smaller than the root element, then move to the left subtree.
4. If it is larger than the root element, then move to the right subtree.

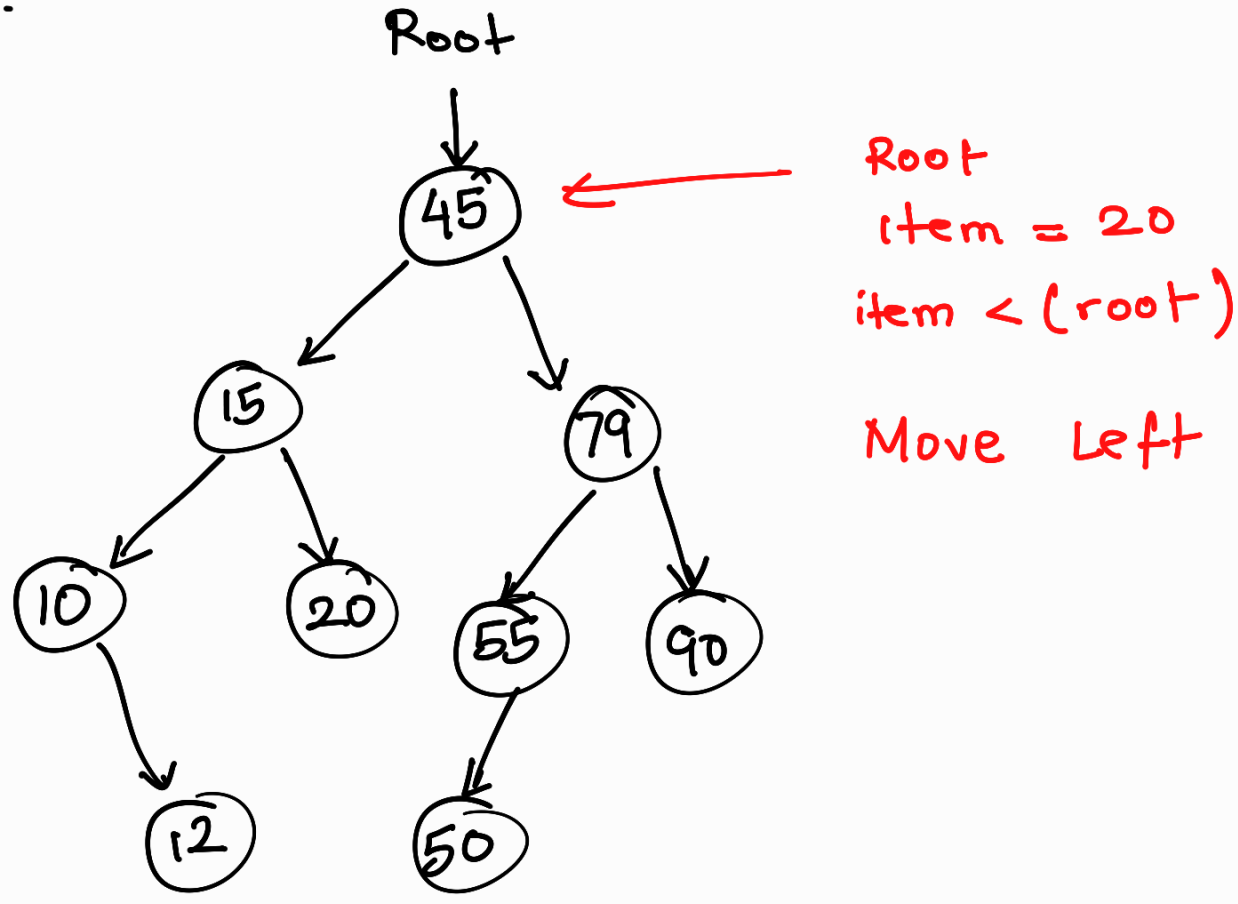
5. Repeat the above procedure recursively until the match is found.

6. If the element is not found or not present in the tree, then return NULL.

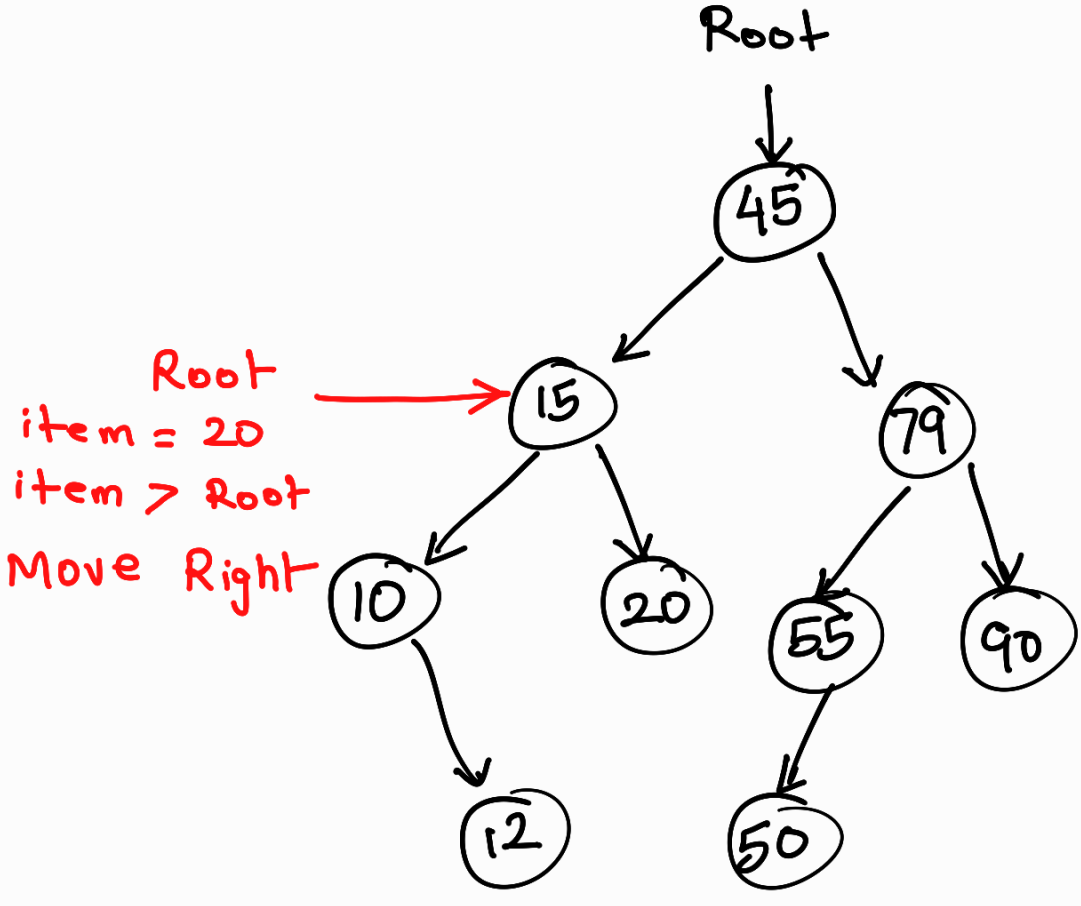


Find 20

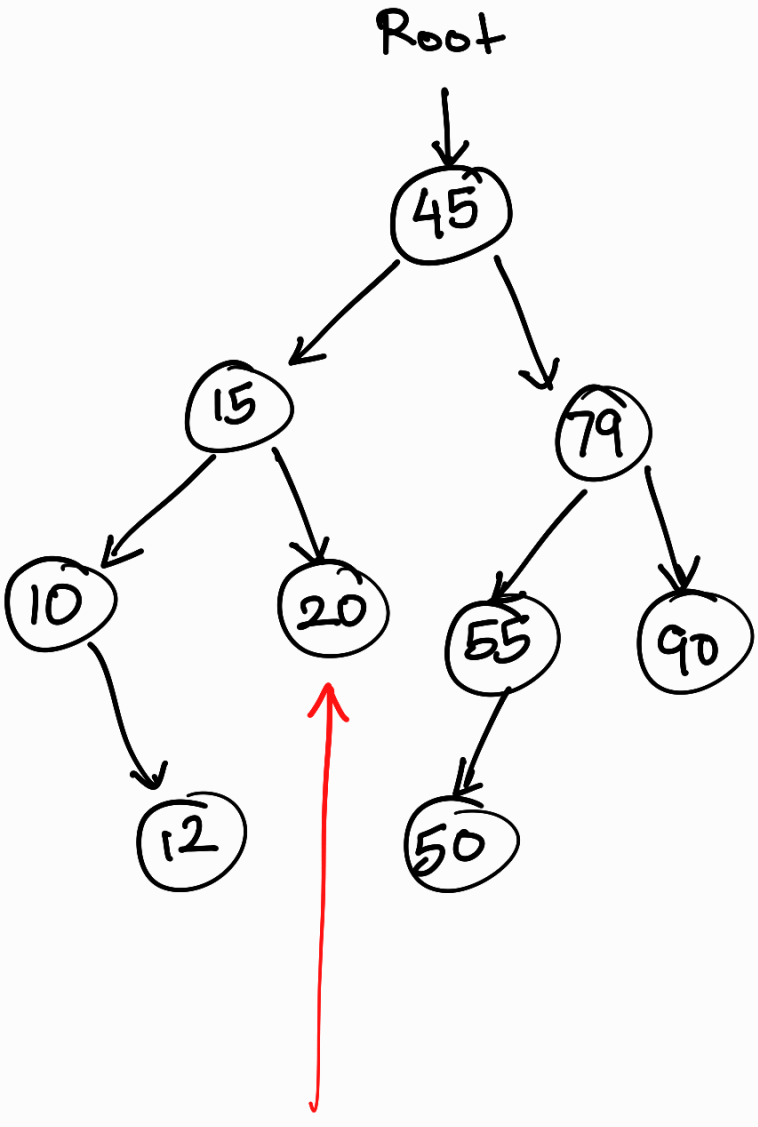
Step 1:


Root
45
15
79
10
20
55
90
12
50
Root
item = 20
item < (root)
Move Left

Step 2:


Root
45
Root
item = 20
item > Root
Move Right
15
79
10
20
55
90
12
50

Step 3 :



Root

item = Root

Return Root

Match Found