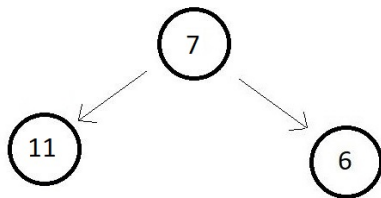


# Binary Search Trees: Introduction & Properties

[codewithharry.com/videos/data-structures-and-algorithms-in-hindi-71](https://codewithharry.com/videos/data-structures-and-algorithms-in-hindi-71)

In the last lecture, we finished learning about all the traversal methods in binary trees. To make it easier for you, we even found a way to keep track of the order of nodes you visit in every traversal method. Today, we'll start learning about what a binary search tree is and what features they possess.

Earlier we concluded a few points about binary trees. Binary trees are normal trees with each parent having either two or less than two children, that is the degree of the tree should be two or less than two.



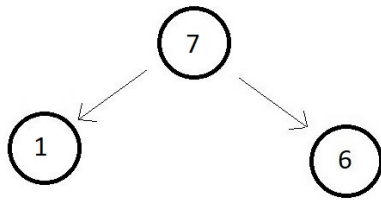
**Figure 1**

Tell me if the above tree is a binary tree or not. Of Course, it is. Since all the nodes have degrees either two or zero. But is this binary tree a binary search tree? Wait. A distinction between binary trees and binary search trees is what I am seeking to establish. Following are the properties of a binary search tree:

## Binary Search Trees:

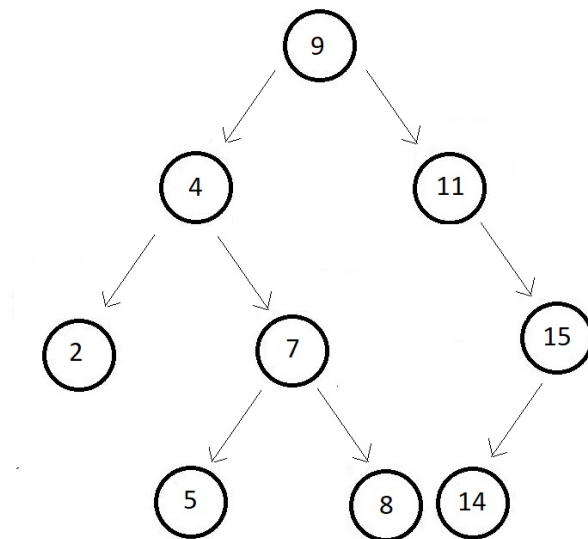
1. It is a type of binary tree.
2. All nodes of the left subtree are lesser than the node itself.
3. All nodes of the right subtree are greater than the node itself.
4. Left and Right subtrees are also binary trees.
5. There are no duplicate nodes.

Having discussed all the properties, you must now tell me if the above binary tree was a binary search tree or not. The answer should be no. Since the left subtree of the root node has a single element that is greater than the root node violating the 2nd property, it is not a binary search tree. Let me take one more example.



**Figure 2**

Tell me if this is a binary search tree. Still no. Because the left subtree is good but the right subtree of the root node is lesser than the root node itself violating the 3rd property. Actually, I have always found you all to be smart enough to answer these petty questions. Let's consider a big one and see if it's a binary search tree or not.



**Figure 3**

Take your time and analyze the different properties of a binary search tree and tell if this is a binary search tree or not.

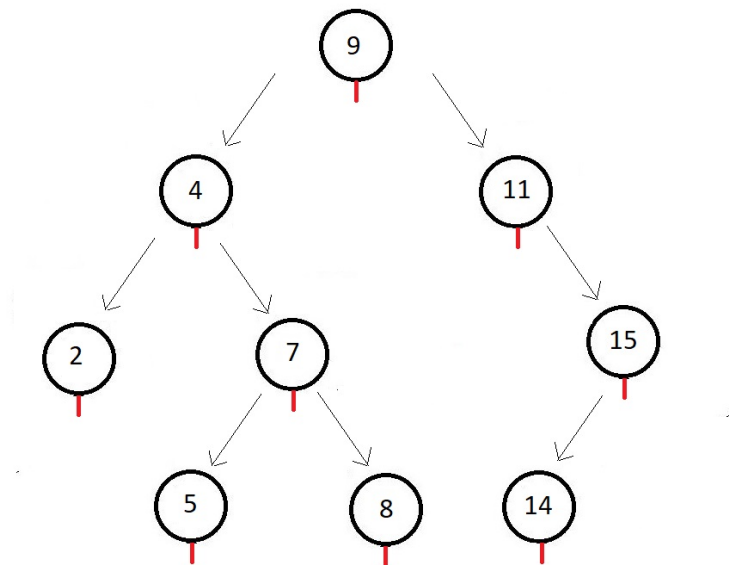
**Solution: YES.**

**Why?**

The very first thing to observe here is the properties of a Binary Search Tree. You would check if all the properties are satisfied for each of the nodes of the tree. So, you first start with the root node which is element 9 and see if all the nodes on the left subtree {4, 2, 5, 7} are smaller than 9 and all the nodes of the right subtree {11, 15, 14} are greater than 9. And since they are, we'll proceed with the next node. Doing this for all the nodes, we'll conclude that this is a Binary Search Tree. Had there been even one violation for any of the nodes, we would have said, no.

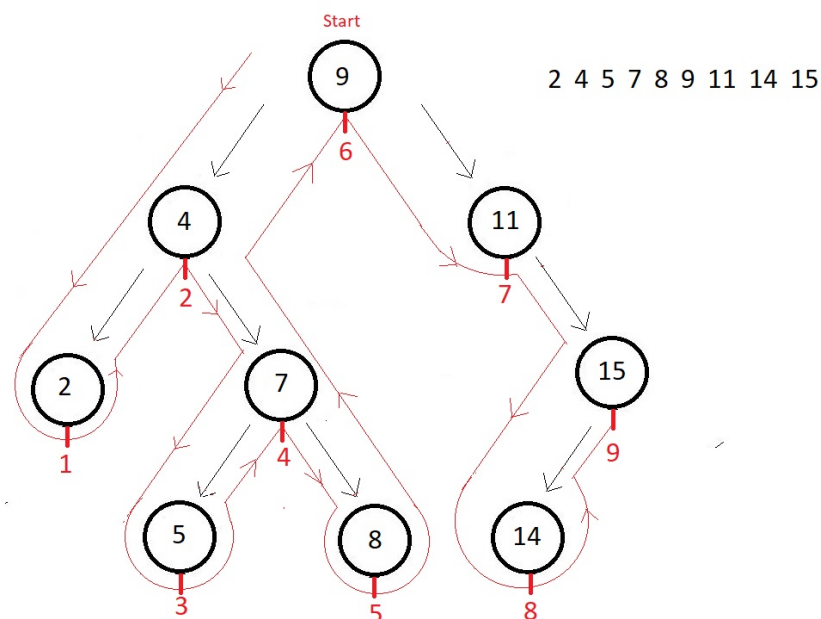
Lastly, there is one more amazing property that I've been keeping all along to amaze you. It says **the InOrder traversal of a binary search tree gives an ascending sorted array**. So, this is one of the easiest ways to check if a tree is a binary search tree. Let's write the InOrder Traversal of the tree in figure 3.

We'll use the same technique we learned in the last lecture to quickly find the InOrder traversal. We'll extend an edge to the bottom of each node.



**Figure 4.**

Now, we'll start from the root node, and drive towards the left subtree and follow the arrows I've drawn for your convenience.



**Figure 5**

So, the final InOrder traversal order of the above tree is

**2 → 4 → 5 → 6 → 7 → 8 → 9 → 11 → 14 → 15,**

which is obviously in increasingly sorted order. Hence, it is a binary search tree.

And this is how easy we have made checking if a tree is a binary search tree or not.

We still haven't seen how to make a binary search tree which shall be our next job. We'll also make programs for creating a binary search tree, and for checking if a tree is a binary search tree or not. This was all about the basics of binary search trees. We introduced this to you and discussed its most important properties.

I appreciate your support throughout. I hope you enjoyed the tutorial. If you genuinely appreciate my work, please let your friends know about this course too. If you haven't checked out the whole playlist yet, move on to [codewithharry.com](https://codewithharry.com) or my YouTube channel to access it. See you all in the next tutorial where we'll write a program to check whether a tree is a binary search tree or not. Till then keep coding.