



## Use Depth First Search in a Binary Search Tree









We know how to search a binary search tree for a specific value. But what if we just want to explore the entire tree? Or what if we don't have an ordered tree and we need to just search for a value? Here we will introduce some tree traversal methods which can be used to explore tree data structures. First up is depth-first search. In depth-first search, a given subtree is explored as deeply as possible before the search continues on to another subtree. There are three ways this can be done: In-order: Begin the search at the left-most node and end at the right-most node. Pre-order: Explore all the roots before the leaves. Post-order: Explore all the leaves before the roots. As you may guess, you may choose different search methods depending on what type of data your tree is storing and what you are looking for. For a binary search tree, an inorder traversal returns the nodes in sorted order.

---

Here we will create these three search methods on our binary search tree. Depth-first search is an inherently recursive operation which continues to explore further subtrees so long as child nodes are present. Once you understand this basic concept, you can simply rearrange the order in which you explore the nodes and subtrees to produce any of the three searches above. For example, in post-order search we would want to recurse all the way to a leaf node before we begin to return any of the nodes themselves, whereas in pre-order search we would want to return the nodes first, and then continue recursing down the tree. Define `inorder`, `preorder`, and `postorder` methods on our tree. Each of these methods should return an array of items which represent the tree traversal. Be sure to return the integer values at each node in the array, not the nodes themselves. Finally, return `null` if the tree is empty.

---

	The <code>BinarySearchTree</code> data structure should exist.
	The binary search tree should have a method called <code>inorder</code> .

	The binary search tree should have a method called <code>preorder</code> .
	The binary search tree should have a method called <code>postorder</code> .
	The <code>inorder</code> method should return an array of the node values that result from an inorder traversal.
	The <code>preorder</code> method should return an array of the node values that result from a preorder traversal.
	The <code>postorder</code> method should return an array of the node values that result from a postorder traversal.
	The <code>inorder</code> method should return <code>null</code> for an empty tree
	The <code>preorder</code> method should return <code>null</code> for an empty tree
	The <code>postorder</code> method should return <code>null</code> for an empty tree