Recall from the learning activity that a Binary Search Tree is a valid Binary Tree where all of the values in the left sub-tree are less than the current node, and all of the values in the right sub-tree are greater than the current node.

1. Describe a recursive function that accepts a valid Binary Tree and determines if it is a Binary Search Tree.

For a recursive function, there must be a base case. The base case for a Binary Search Tree function is that if the node is null, this means that all the recursive calls are completed, and the tree is a Binary Search Tree. The function will return true. As we know for a binary tree, the left child node of a parent node must have a smaller value than the parent node, while the right node must have a greater value than the parent node. If any of the above conditions are not fulfilled, the Binary Tree is not a Binary Search Tree. The function will return false.

For the recursive calling part of the function, first, it will check the left subtree and traverse through all the child nodes, followed by the right subtree, starting from the root node. If neither true nor false is returned, meaning the left child node is smaller in value than its parent node, and the right child node is greater in value than its parent node, or it still has not yet reached the end of the Binary Tree, it calls itself again, passing the left child node as a current node and keeps traversing its child nodes, the return statement also includes a call with the right child node, similar to the left child node, until it reaches a null node, which implies the end of the Binary Tree, and will return true, otherwise, it will return false.