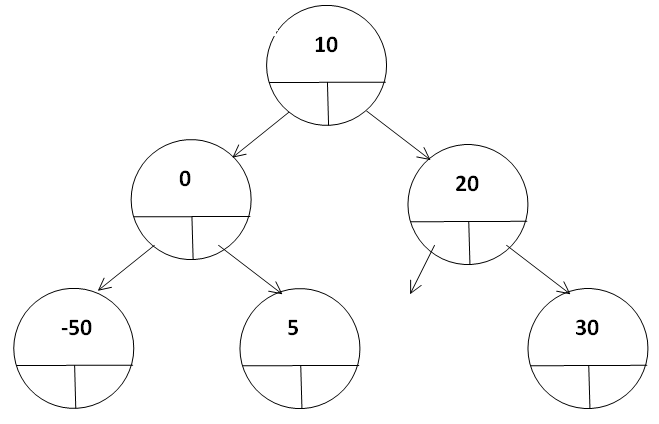
## **Find and Remove Next Smaller Value**

**The goal for this exercise** is to implement the BST.FindAndRemoveNextSmallerValue method, as a first step towards implementing the overall BST.Remove method.

Implement the FindAndRemoveNextSmallerValue method. This method is given a target number and a node to start it’s search at. The node to start searching at contains the target number (yes, the two parameters are redundant ☺). The method should then proceed to find (and remove) the BST Node that contains the value that is the next smaller value than the target value. The return value is a reference to the node that you just removed.

For example, given the following tree:



You might be given the number ‘10’, and a reference to the root node. In that case, you would need to find the next smaller number that’s in the tree (which is 5), and remove it. You would then return a reference to that node to the caller.

Another example: you might be given the number ‘0’, and a reference to the node that is the left child of the root node. In that case, you would need to find the next smaller number that’s in the tree (which is -50), and remove it. You would then return a reference to that node to the caller.

**What you need to do for this exercise:**

1. In the provided starter solution, there is a project named **PCE\_ForTests**, there is a class named BinarySearchTree (in the file named PCE\_09.cs)**.** For this exercise, you should implement the ‘FindAndRemoveNextSmallerValue method, as described in the lecture.
2. Once you’ve done this, all the tests in the BST\_FindAndRemoveNextSmaller class should pass.   
   1. Notice that since the method you need to implement is private, the method will be tested with the help of an additional, public method that we’ve added to the BST: TestFindAndRemoveNextSmallest. This method takes a given value, finds the node containing that value in the tree, and then calls FindAndRemoveNextSmallerValue on that node.