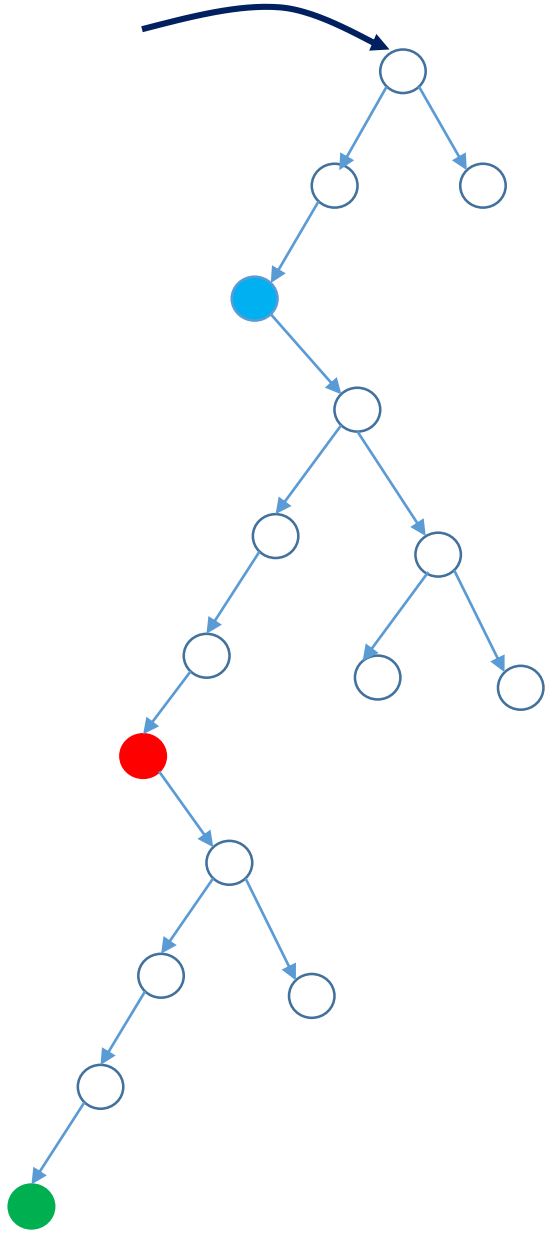

In the BST shown, the blue node has the smallest value in the tree; the red node has the second smallest value, and the green node the third smallest value.

The external iterator should maintain a (private) stack of nodes, with the invariant that the next value to be returned is in the node pointed to by the top of the stack. Therefore, when the external iterator is first created and is given a BST such as shown on the left, it should traverse the left spine of the BST and stack all nodes so that the blue node is at the top of the stack.

When the next() method is invoked, the value to be returned is at the top of the stack. The stack is then popped, and, if the right subtree of the popped node is not null, the left spine of the right subtree is traversed and all nodes are stacked, as before. In the example shown, this will result in the red node getting to the top of the stack.

This is the overall idea; the remaining details are to be filled in by you.