Exercise 3.2.36: Iterator. Is it possible to write a nonrecursive version of keys() that uses space proportional to the tree height (independent of the number of keys in the range)?

Traversals note:

- inorder traveral: Traverse left subtree, root, right subtree
- preorder traversal: visit root, left subtree, right subtree
- postorder traversal: left, right, root
- level-order traversal: all nodes on same level. Used in BFS traversals.

Solution: Yes, it is possible to write a nonrecursive solution. However, this will require a stack.

1. Take the root of the tree and all of its left children onto the stack.
2. Check is stack empty. If not empty, then pop top element of the stack. Print it, and replace its current position on stack with elements right child.
3. Add the right child and all its left children to the stack.
4. Repeat steps 2 AND 3 until stack is empty
5. Notice how we are pushing all left children to the stack, popping top elements off and then pushing its respective right children nodes onto the stack to replace its current position. This resembles inorder traversal.

Summary: For range [lo..hi], simply search for lo and hi in the tree, placing a new pointer to the next node in each node of the (two) search paths. Note that you need extra space for the pointers at most twice the height of the tree (for marking the two paths). Then do an (iterative) inorder traversal of the BST, using the extra pointers to not visit parts of it that do not belong to the range.