10.4-4.

Write an $O(n)$-time procedure that prints all the keys of an arbitrary rooted tree with n nodes, where the tree is stored using the left-child, right-sibling representation.

Answer.

As Figure 10.10 shows, the left-child, right-sibling representation make it easy to print the tree in a recursive way.

```
PRINT-LCRS-TREE(T)
 1  x = T.root
 2  if x ≠ NIL
 3       print x.key
 4       lc = x.left-child
 5       if lc ≠ NIL
 6            PRINT-LCRS-TREE(lc)
 7            rs = lc.right-sibling
 8            while rs ≠ NIL
 9                 PRINT-LCRS-TREE(rs)
10                 rs = rs.right-sibling
```

We can see that to print nodes in a nonempty tree, the PRINT-LCRS-TREE procedure first prints the node at its root, it then apply itself to all the root's children (which might also be nonempty trees) sequentially. The step required by this procedure is proportional to the amount of nodes in the tree, which is $\Theta(n)$. Hence, it takes PRINT-LCRS-TREE $O(n)$ time to print all the keys of an arbitrary rooted tree with n nodes.

Email address: informlarry@gmail.com