

Tree Iteration, Reading

"Iteration" is a loop process that reaches every value in a data structure. We've called it "traversal" in reference to arrays and linked lists. For linear structures, it's a simple for-loop:

```
for (i = 0; i < n; i++)
    ...a[i]...
```

```
for (p = firstNode; p; p = p->next)
    ...p->value...
```

```
for (queue<int> keys = a.keys(); !keys.empty(); keys.pop())
    ...keys.front()...
```

For tree structures, there are *many paths* from the top to the bottom, so a simple loop won't work. In this module we came up with recursive solutions that looked like these:

```
clear(p->left);
clear(p->right);
delete p;
```

```
keys(p->left, q);
q.push(p->key);
keys(p->right, q);
```

```
p->value = copiedTree->value
copyTree(p->left, copiedTree->left)
copyTree(p->right, copiedTree->right)
```

What each of these have in common is that there is one statement for dealing with the currently pointed-to node (using a "p" pointer) and a pair of statements to start the next cycle of the loop for each of its children. What's *different* is *when* in the order of things the "p" node itself is handled. When deallocating a tree, the "p" node cannot be deallocated until its descendants are. In queuing up keys, the "p" node has to be included *between* its left and right children if we want the keys in sorted order. And in copying another tree, the "p" node has to exist before its children are added. Order matters.

Tree Terminology

In tree technology, the node pointed to by the traversing "p" pointer is called the **"vertex"**, abbreviated **V**. The left and right descendants are abbreviated **L** and **R**. Using these abbreviations, there are ways to refer to the three ways to order them in a recursive process:

- LRV, "postorder" that was applied in deallocation
- LVR, "inorder" that was applied in our keys function
- VLR, "preorder" that was applied in copying another tree

Of course, there are the *compliments* of these with R before L, but whichever order for L and R that the programmer wishes to use, it's the placement of V that's important, and hence the naming -- pre, in, and post -- refer to the vertex.