## C-6.16

With an array-based implementation, the binary tree functions positions() and elements() each take $O(n)$ time. This is because we need to step through the entire array and extract each position/element. The swap and replace functions take constant time because of the fact that we can access elements in the array using an index, without having to search the entire array. The root(), parent(), children(), leftChild(), rightChild(), and sibling() functions are also constant because we can find these items with simple calculations (for example, the right child of a node with index $n$ is $n+2$). Finally, isInternal(), isExternal(), and isRoot() also only take a quick calculation based on indices (whether an index is in the last half of the array, is the first item in the array, etc.). Thus, these functions are also constant.