15.1-5 Given an element \( x \) in an \( n \)-node order-statistic binary tree and a natural number \( i \), how can the \( i \)th successor of \( x \) be determined in \( O(\lg n) \) time.

This problem can be solved if our data structure supports two operations:

- **Rank(\( x \))** – what is the position of \( x \) in the total order of keys?

- **Get(\( i \))** – what is the key in the \( i \)th position of the total order of keys?

What we are interested in is \( \text{Get}(\text{Rank}(x) + i) \).

In an order statistic tree, each node \( x \) is labeled with the number of nodes contained in the subtree rooted in \( x \).

\[ \text{Implementing both operations involves keeping track of how many nodes lie to the left of our path.} \]