Question 4.5 from CTCI: Find the Next Successor

September 7, 2017

Question

Write an algorithm to find the next node (e.g., in-order successor) of a given node in a binary search tree where each node has a link to its parent. You will write a function which takes a TreeNode e and returns a TreeNode p which is the successor of e.

Explanation and Algorithm

We approach this problem by thinking very, very carefully about what happens on an inorder traversal. On an in-order traversal, we visit X.left, then X, then X.right. So, if we want to find X.successor(), we do the following:

1. If X has a right child, then the successor must be on the right side of X (because of the order in which we visit nodes). Specifically, the left-most child must be the first node visited in that subtree.
2. Else, we go to Xs parent (call it P).
    - If X was a left child (P.left = X), then P is the successor of X
    - If X was a right child (P.right = X), then we have fully visited P, so we call successor(P).

Hints

1. Think about how in-ordertraversal works. What are the possible cases of the given node; is it a left child or right child of its parent node?
2. What should we do if the given node is a right child? If it's a left child?

Code

```
public static TreeNode inorderSucc(TreeNode e) {
   if (e != null) {
      TreeNode p;
 // Found right children -> return 1st inorder node on right
      if (e.parent == null || e.right != null) {
         p = leftMostChild(e.right);
      } else {
 // Go up until  were  on left instead of right (case 2b)
         while ((p = e.parent) != null) {
            if (p.left == e) {
               break;
            }
            e = p;
         }
      }
      return p;
      }
   return null;
 }

 public static TreeNode leftMostChild(TreeNode e) {
   if (e == null) return null;
      while (e.left != null) e = e.left;
   return e;
 }
```

Big O analysis

The time complexity of this algorithm is $O(h)$, where h is the height of the binary search tree. This is because we will, at worst, have to iterate through each "level" of the tree until we reach the bottom/left-most child node.

Sources

Question, answer and other material taken from Cracking the Coding Interview 6th edition by Gayle Laakmann McDowell.