Programming Interviews Exposed

Secrets to Landing Your Next Job

Duplicate Values in Binary Trees

By Eric Giguere

Like 5 Tweet G+1 1 Share

Every once in a while I get an email from a reader pointing out a possible error in the book. There are definitely some errors in there. But sometimes they aren't errors, but I treat them as such because there's obviously something that needs to be clarified.

Here's a recent email from a reader (hi Mark!) that warrants extra discussion:

> *Anoth rror I spotted is that binary search trees usually do not contain duplicate nodes ences at the top of page 64 should say that the nodes to the left are all less than the nc ue and the nodes to the right are all greater than the nodes own value. But, if duplic ere allowed, then either the left or the right would have the "or equal to" but not both (vording has both sides having "or equal to.".*

5 Like Tweet 1 G+1

It's definite most binary search trees (BSTs) you see in examples don't contain nodes with duplicate k er, that doesn't mean it isn't possible. If you can only store one value in a node and you ha values that map to the same key then you'll need multiple nodes with duplicate keys to rep hose values. (In real life, a BST will have a value associated with it, say a pointer to a struct amples in books and interviews won't add the extra complexity of value manageme there in practice.)

What Mark is objecting to is the "less than or equal to" and "greater than or equal to" wording when describing how the nodes of a BST relate to their parents. He's asserting that either:

1. The left child must be less than or equal to the parent and the right child must be greater than the parent; or
2. The left child must be less than the parent and the right child is greater than or equal to the parent.

Enforcing either statement will still give you a BST. But it will make it impossible to create a *balanced* tree. (Recall that in a balanced tree the difference in heights of the left and right subtrees of a node is either 0 or 1.)

Let's say you have a small tree of 7 nodes, each with value "5". Create a tree from them using either of Mark's rules and what do you end up with? A tree with nothing but left or right children — effectively a linked list. The tree has a height of 7.

If you relax the rules to allow for "less than or equal to" *and* "great than or equal to" then you can create a balanced tree of height 3. Balanced trees are great because you can search them in O(log n) time.

Of course, all of this hinges on allowing duplicate keys in a BST. If you don't allow duplicate keys then the point is moot since "less than or equal to" and "greater than or equal to" devolve to "less than" and "greater than". This is the likeliest case, in fact, since most tree implementations I've seen treat the keys as a set, not just a collection.

Anyhow, obviously something that could use some clarification in the book. Thanks Mark!

Feel free to send questions and comments about anything you've read in the book to eric@piexposed.com. Sometimes I'm slow with my mail, so if you don't get a response after a couple of days, resend it to remind me to reply…

Share and Enjoy

If you liked this, please share it with others using the buttons below!

This entry was posted in General on October 18, 2013 [http://www.piexposed.com/duplicate-values-in-binary-trees/] .