Lab 5: 2-3 Trees

Week of November 28, 2016

Objective

To complete a small program that builds 2-3 trees and searches in them.

The Program

File `Lab5.java` contains a nearly-complete program that constructs a number of 2-3 trees and searching in each one. You will complete the program by writing the body of the `private` helper method `findLeaf` (see "Exercise" below).

The program creates a number of 2-3 trees. For each tree, it randomly chooses

- The number of insertions to perform, and
- The values to insert.

The insertion method does not permit duplicates, so the number of values actually inserted may differ from the number of insertions that was randomly chosen.

For each of the 2-3 trees created, it first prints out the tree after the tree is created using an inorder traversal, so you can **see** what's in the 2-3 tree. After creating the tree, it does two batches of searches:

- It searches for values that were inserted into the tree (it stores inserted values in an array, so that it randomly chooses an index in the array to tell it what value to search for). These searches should all succeed.
- It then searches for random values. Since they are randomly-chosen values, most of them are not in the tree, so most of these searches will fail.

Exercise

Your task is to write the body of `findLeaf` in the `TwoThreeTree` class. You will find it at the very end of the file. Method `findLeaf` is passed a data item, `searchItem`. This method's task is to find and return a pointer to L_{search}, the leaf where a search for `searchItem` ends.

You can write this method in one of two ways:

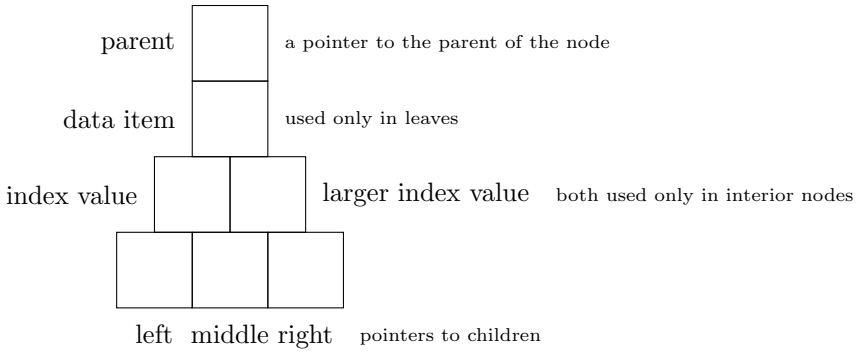
- Iteratively — that is, with a loop, moving down the tree until a leaf is reached; or
- Recursively. If you use recursion, you will need to add a recursive helper method in the `TwoThreeNode` class to do the work and `findLeaf` in the `TwoThreeTree` class will simply be the driver method.

To write this method, you need to understand the nodes (see below).

Just add the body of the `findLeaf` (and the `TwoThreeNode` class recursive helper method if you decide to do the search recursively), and change nothing else in the `Lab5.java` file.

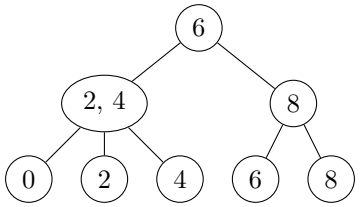
The Leaf-based 2-3 Tree Node Implementation

A 2-3 tree node is represented by an instance of the `TwoThreeNode` class. A `TwoThreeNode` has a lot of instance variables:

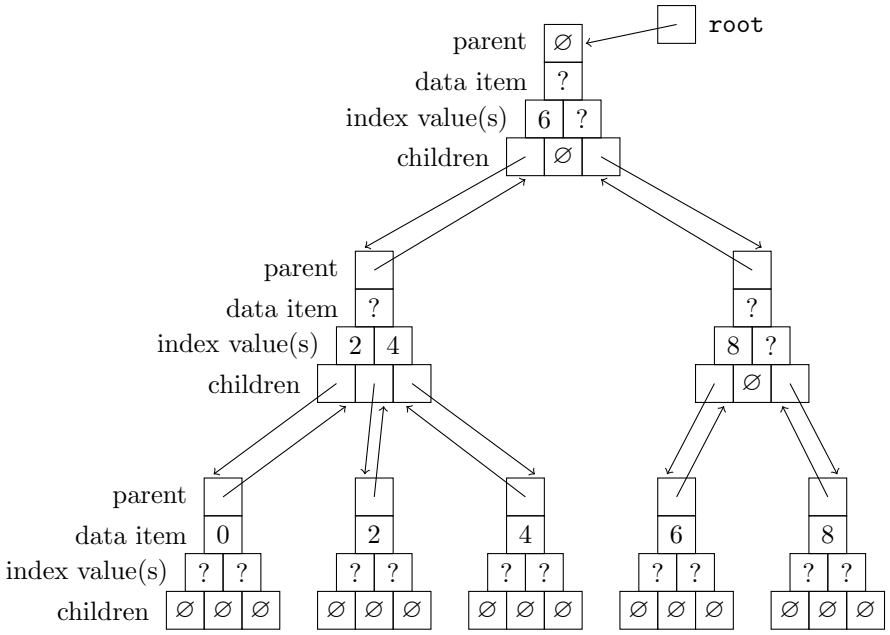


Remember that data items are stored only in the leaves. An interior node does not contain a data item, only one or two index values to guide searches to the correct leaf (the number of index values depends on the number of children).

For example, consider the following leaf-based 2-3 tree:



Here's how it is implemented using the `TwoThreeNode` class:



Notice that a leaf contains a data item, the index values are not used in the leaf, and the children are all `null` in the leaf. On the other hand, in an interior node, no data item is stored; instead, an interior node contains one or two index value (depending on whether it has two or three children, respectively). Notice also that an interior node with two children does not use the middle child pointer (the middle child pointer is `null`).