Best Trick To Find PreOrder, InOrder & PostOrder Traversal

Show Course Contents

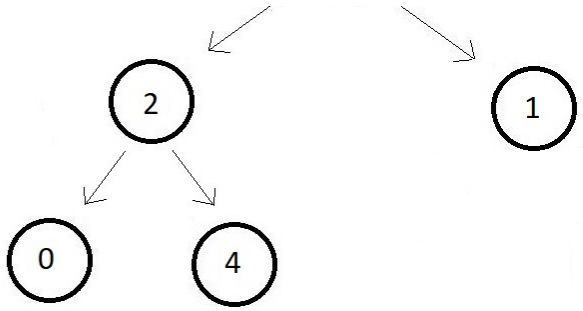
Overview Q&A Downloads Announcements

Best Trick To Find PreOrder, InOrder & PostOrder Traversal

In the previous tutorial, we covered our last traversal technique, the InOrder Traversal. We dealt with its programming in C. Earlier to that, we saw the other two traversal techniques. In the end, I did tell you that we will pursue a faster method to traverse through a Binary Tree. So today we'll see a faster, in fact, the best trick to find the PreOrder, the InOrder, and the PostOrder Traversal in binary trees.

This method shall help you in case you are in a hurry, or you have an assessment the next day. If it's just about marks, you can count entirely on the lecture and get good grades, I promise you. Throughout this lecture, it is assumed you have jumped right into this lecture, and haven't gone to the previous lectures of the series.

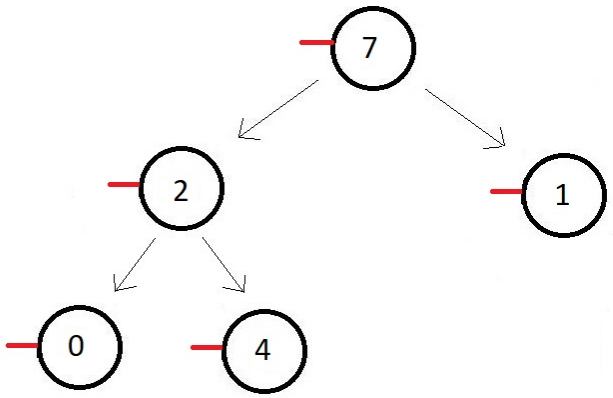
I would like to provide an example of a Binary Tree. We will then traverse through it using different techniques.



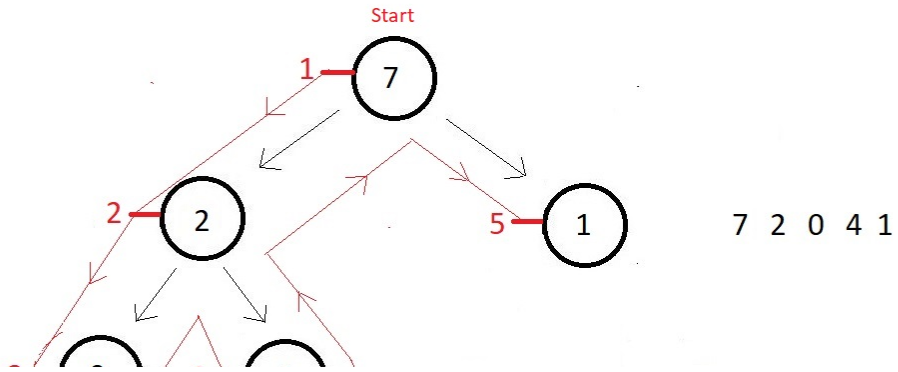
First, we'll see traversal using the PreOrder approach.

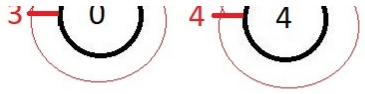
PreOrder Traversal:

In this method, we start with the node, and then move to the left subtree and then to the right subtree. But the trick says, extend an edge to the left of all the nodes. Follow the figure below to understand what is being said.



And now, start driving from the root to the left of the root node, and whenever you intersect a red edge while driving, print its value. Refer to the illustration below. Arrows have been made to direct you to the path we have followed.

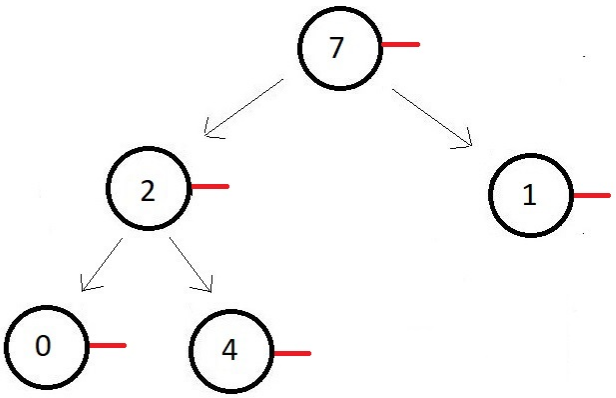




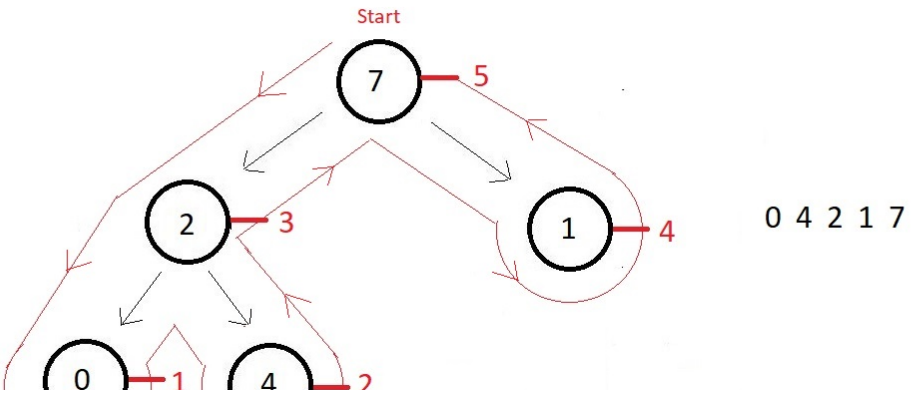
And this is how you were quickly able to write the order of nodes you visited first in the PreOrder Traversal. It was 7→2→0→4→1. Let's see how easy PostOrder Traversal is.

PostOrder Traversal:

In this method, we start with the left subtree first, and then move to the right subtree, and then finally to the root node. But the trick illustrates, extending an edge to the right of all the nodes. Earlier it was left, now it is right. Follow the figure below to understand.



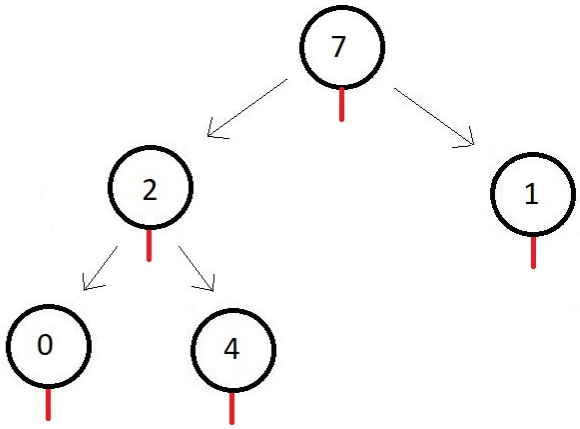
And now, drive the same way we did earlier. Start driving from the root to the left of the root node and whenever you intersect a red edge while driving, print its value. Refer to the illustration below. Arrows have been drawn to direct you to the path we have taken.



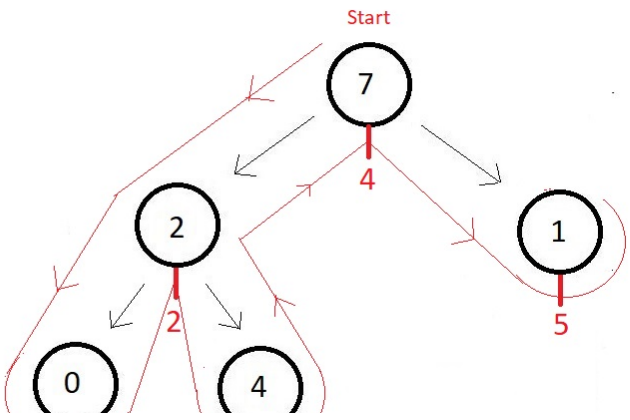
And this is how you were quickly able to write the order of nodes you visited first in the PostOrder Traversal. It was 0→4→2→1→7. Let's see lastly how easy InOrder Traversal is.

InOrder Traversal:

In this method, we start with the left subtree first, and then move to the root node and then finally to the right subtree. But the trick says, extend an edge to the bottom of all the nodes. We went to the left, to the right, and now to the bottom. Follow the figure below to understand.



Then, drive as we have done all along. Start driving from the root to the left of the root node and whenever you intersect a red edge while driving, print its value. Refer to the illustration below. Arrows have been drawn to direct you to the path we have taken.



0 2 4 7 1

And this is how you were quickly able to write the order of nodes you visited first in the InOrder Traversal. It was 0→2→4→7→1.

You can do one thing now. Create the binary tree in the above example and use the programs we created in earlier lectures for each of these traversals to verify if this trick actually works. And I can assure you the authenticity of this trick for all the larger binary trees you will get in your assessments. Use this blindly. But make sure you cover up the concepts sooner or later. Because that is what helps you in the end.
Here we mark the end of our traversal techniques. Next on the list is a much-anticipated topic, Binary Search Trees (BST). I hope you'll join me there.

I appreciate your support throughout. I hope you enjoyed the tutorial. If you genuinely appreciate my work, please let your friends know about this course too. If you haven't checked out the whole playlist yet, move on to codewithharry.com or my YouTube channel to access it. See you all in the next tutorial where we'll start with one of the most important topics, the **Binary Search Trees**. Till then keep coding.

Previous Next