Tree Traversal

When traversing a binary tree, there are three ways that it can be done: pre-order, in-order and post-order.

**Pre-order Traversal:**

It is known as the top down traversal method and it encounters all the roots before it encounters all of the leaves of the tree. The recursive algorithm that can be used to traverse a binary tree in a pre-order way is:

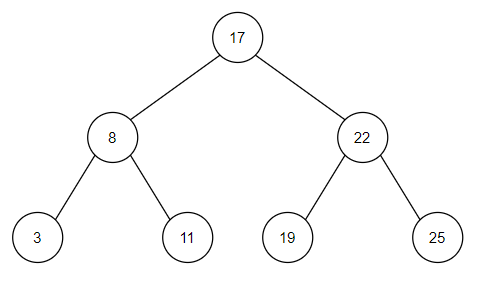
PreorderTraverse (Tree)

Visit the root of the tree

PreorderTraverse (left subtree)

PreorderTraverse (right subtree)

This algorithm must be remembered for the exam however, there is an easier way to know how to traverse the tree without having to go through the algorithm like a computer would have to.

Take this tree:







You could think about traversing this tree by drawing an outline around the tree as shown above. You then put a pointer every time you pass the left of a node. You then output the answer in the order than your points are which gives the pre-order traversal of that tree, in this case “17 8 3 11 22 19 25”.

A pre-order traversal can be used to produce a prefix notation which is used in functional programming. An example of this is writing A+B (infix) as +AB (prefix).

**In-order Traversal:**

This traversal visits the node in sequential order. The recursive algorithm that can be used to traverse a binary tree in an in-order way is:

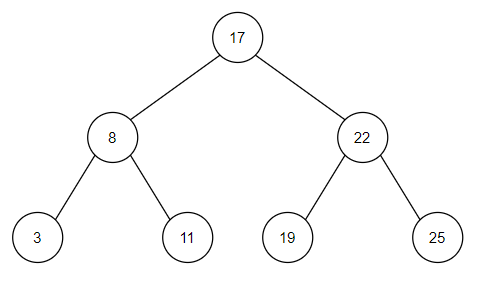
InorderTraverse (Tree)

InorderTraverse (left subtree)

Visit the root of the tree

InorderTraverse (right subtree)

This algorithm must be remembered for the exam however, there is an easier way to know how to traverse the tree without having to go through the algorithm like a computer would have to.

Take this tree:







You could think about traversing this tree by drawing an outline around the tree as shown above. You then put a pointer every time you pass the bottom of a node. You then output the answer in the order than your points are which gives the in-order traversal of that tree, in this case “3 8 11 17 19 22 25”.

An in-order traversal can be used to alphabetically sort the order of name or numerically sort the order of numbers of the tree. In this example the in-order traversal outputs the answer in order of ascending values. If they were names it would output the answer in order of which comes first in the alphabet.

**Post-order Traversal:**

It is known as the bottom down traversal method and it encounters all the leaves before it encounters all of the roots of the tree. The recursive algorithm that can be used to traverse a binary tree in a post-order way is:

PostorderTraverse (Tree)

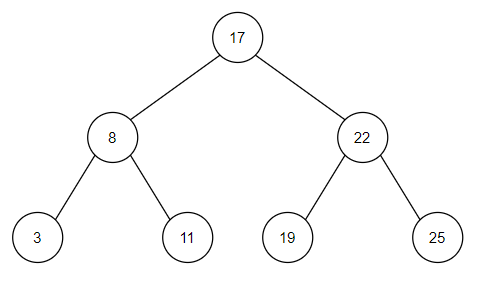
PostorderTraverse (left subtree)

PostorderTraverse (right subtree)

Visit the root of the tree

This algorithm must be remembered for the exam however, there is an easier way to know how to traverse the tree without having to go through the algorithm like a computer would have to.

Take this tree:







You could think about traversing this tree by drawing an outline around the tree as shown above. You then put a pointer every time you pass the right of a node. You then output the answer in the order than your points are which gives the post-order traversal of that tree, in this case “3 11 8 19 25 22 17”.

A post-order traversal can be used to produce a postfix notation which is used in functional programming. An example of this is writing A+B (infix) as AB+ (postfix – known as Reverse Polish Notation RPN).