DEPTH FIRST TREE TRAVERSAL

The Algorithm of discussion is pre-order transversal algorithm

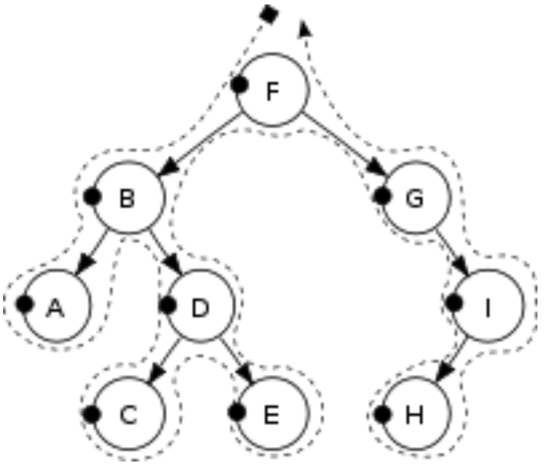
Pre-order transversal algorithm:
The algorithm can be written as:

```
Sub P(TreeNode)

    Output(TreeNode.value)
    If  LeftPointer(TreeNode) != NULL Then
        P(TreeNode.LeftNode)
    If RightPointer(TreeNode) != NULL Then
        P(TreeNode.RightNode)

End Sub
```

This algorithm can be summed up as:

1. Visit the root node (generally output this)
2. Traverse to left subtree
3. Traverse to right subtree

Given this tree below:



The algorithm will output: F,B,A,D,C,E,G,I,H

This algorithm is used to create a copy of a tree or domain
It is basically useful when functions depend on others on those functions must be performed first before those that depend on them.

This algorithm however is not preferred to any other depth first tree transversal algorithm, as you already know, Picking an algorithm therefore depends on the domain and its requirements.