Trees Notes

1) Trees can have the Nodes and the edges

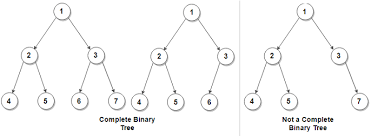
where edges = node -1

trees can have different shapes and sizes

- Complete binary tree vs Full tree

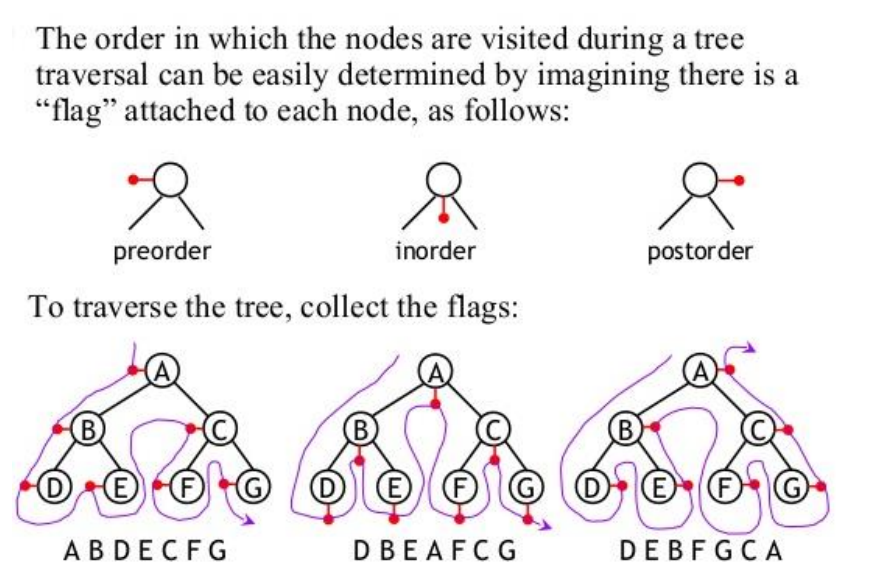
- Full tree - every node in the tree either have 0 or 2 children

- Complete tree - if you stack the tree in the array from L -> R then the array that been created shouldn’t have any spaces in between



Tree Traversal:

Three methods are there to traverse the tree

1. Pre Order: Node -> Left child -> Right Child
2. In Order: Left child -> Node -> Right Child
3. Post order: Left -> Right -> Node
4. 

Heap: It’s a pile or a stack of the records

In order Tree Traversal without Recursion

Using [Stack](http://en.wikipedia.org/wiki/Stack_%28data_structure%29)is the obvious way to traverse tree without recursion. Below is an algorithm for traversing binary tree using stack. See [this](http://neural.cs.nthu.edu.tw/jang/courses/cs2351/slide/animation/Iterative%20Inorder%20Traversal.pps)for step wise step execution of the algorithm.

1) Create an empty stack S.

2) Initialize current node as root

3) Push the current node to S and set current = current->left until current is NULL

4) If current is NULL and stack is not empty then

a) Pop the top item from stack.

b) Print the popped item, set current = popped\_item->right

c) Go to step 3.

5) If current is NULL and stack is empty then we are done.

# Iterative Preorder Traversal

Given a Binary Tree, write an iterative function to print Preorder traversal of the given binary tree.

Refer [this](https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder/)for recursive preorder traversal of Binary Tree. To convert an inherently recursive procedures to iterative, we need an explicit stack. Following is a simple stack based iterative process to print Preorder traversal.  
**1)** Create an empty stack nodeStack and push root node to stack.  
**2)** Do following while nodeStackis not empty.  
….**a)** Pop an item from stack and print it.  
….**b)** Push right child of popped item to stack  
….**c)** Push left child of popped item to stack

Right child is pushed before left child to make sure that left subtree is processed first.

# Iterative Post Order Traversal

# https://www.geeksforgeeks.org/iterative-postorder-traversal/

1. Push root to first stack.

2. Loop while first stack is not empty

2.1 Pop a node from first stack and push it to second stack

2.2 Push left and right children of the popped node to first stack

3. Print contents of second stack