Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

YEAR II SEM III

CS8391

DATA STRUCTURES (COMMON TO CSE &IT)

UNIT -III

NON LINEAR DATA STRUCTURES
TREES
3.4 Threaded Binary Tree

4 QUALITY EDUCATION

NBA NATIONAL BOARD OF ACCREDITATION

NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

nirf NATIONAL INSTITUTIONAL RANKING FRAMEWORK

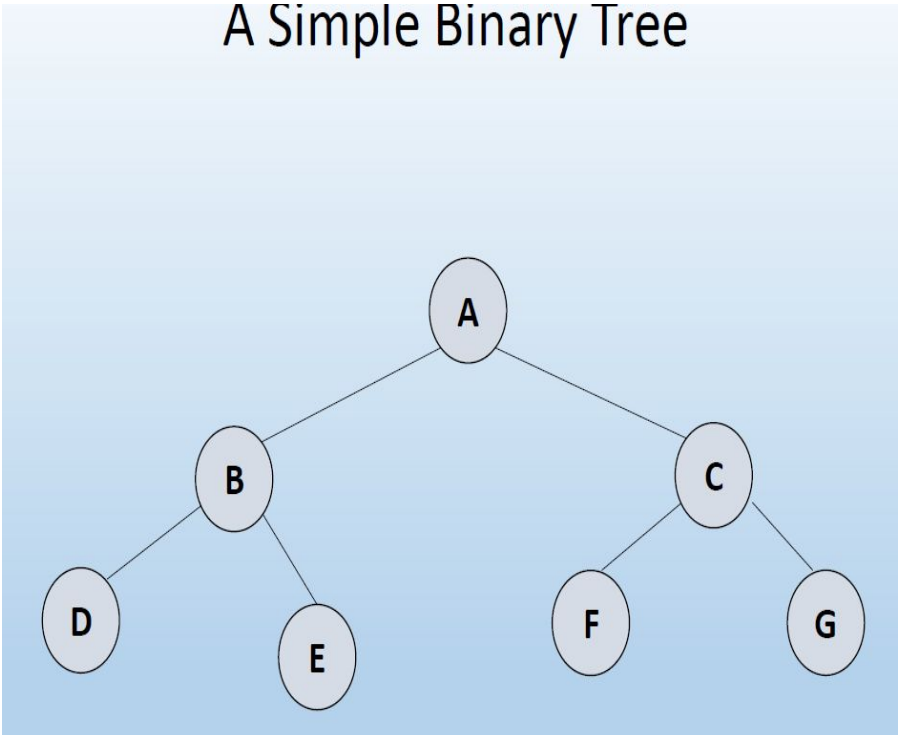
ISO 9001 : 2015 BUREAU VERITAS Certification

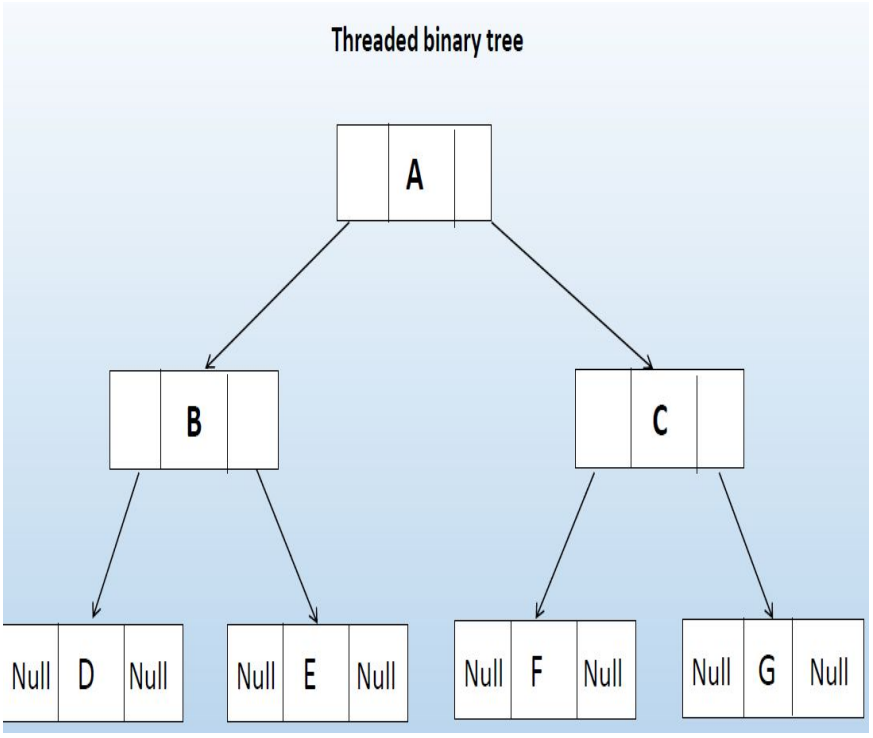
THREADED BINARY TREE

A binary tree where the memory of nodes will be wasted by not linking to any node, such a kind of memory will be used to store the inorder successor of the node if exists. These kind of trees are called as threaded binary trees.Threaded Binary Tree is also a binary tree in which all left child pointers that are NULL (in Linked list representation) points to its in-order predecessor, and all right child pointers that are NULL (in Linked list representation) points to its in-order successor.

Perlis and C. Thornton have proposed new binary tree called "***Threaded Binary Tree***", which makes use of NULL pointers to improve its traversal process. In a threaded binary tree, NULL pointers are replaced by references of other nodes in the tree. These extra references are called as ***threads***.

If there is no in-order predecessor or in-order successor, then it points to the root node.The idea of threaded binary trees is to make inorder traversal faster and do it without stack and without recursion.





In Threaded binary tree, there are 8 null pointers & actual 6 pointers. In all there are 14 pointers.We can generalize it that for any binary tree with n nodes,there will be (n+1) null pointers and 2n total pointers.

The objective here to make effective use of these null pointers.Proposed idea to make effective use of these null pointers.According to this idea we are going to replace all the null pointers by the appropriate pointer values called threads.

And binary tree with such pointers are called **threaded tree.**

In the memory representation of a threaded binary tree, it is necessary to distinguish between a normal pointer and a thread.

2 types -

1.Single Threaded / one-way threading

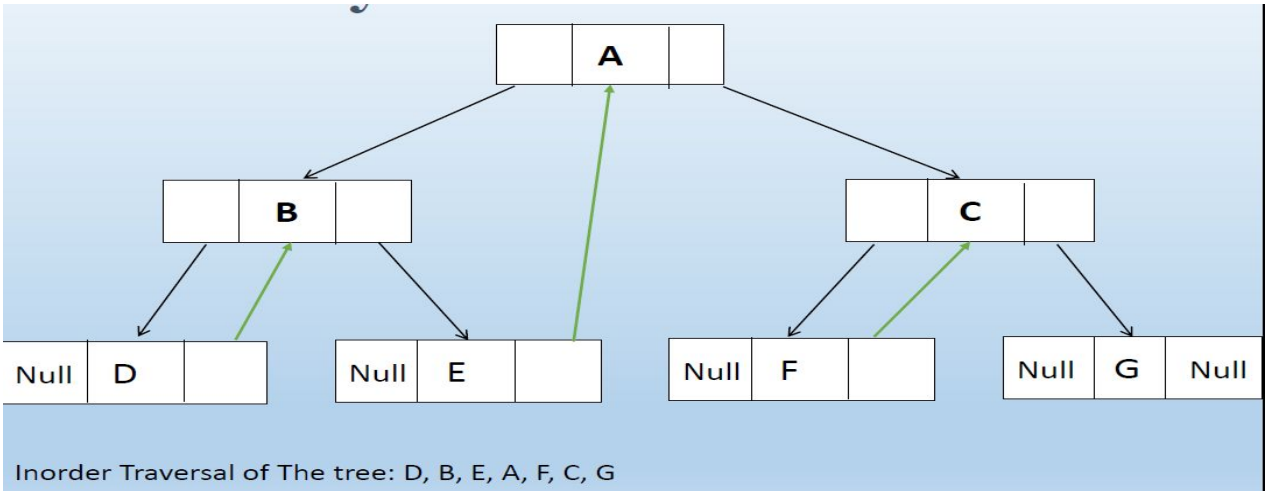
- Where a NULL right pointers is made to point to the inorder successor (if successor exists)

2.Double threaded / two-way threading

- Where both left and right NULL pointers are made to point to inorder predecessor and inorder successor respectively. The predecessor threads are useful for reverse inorder traversal and postorder traversal.

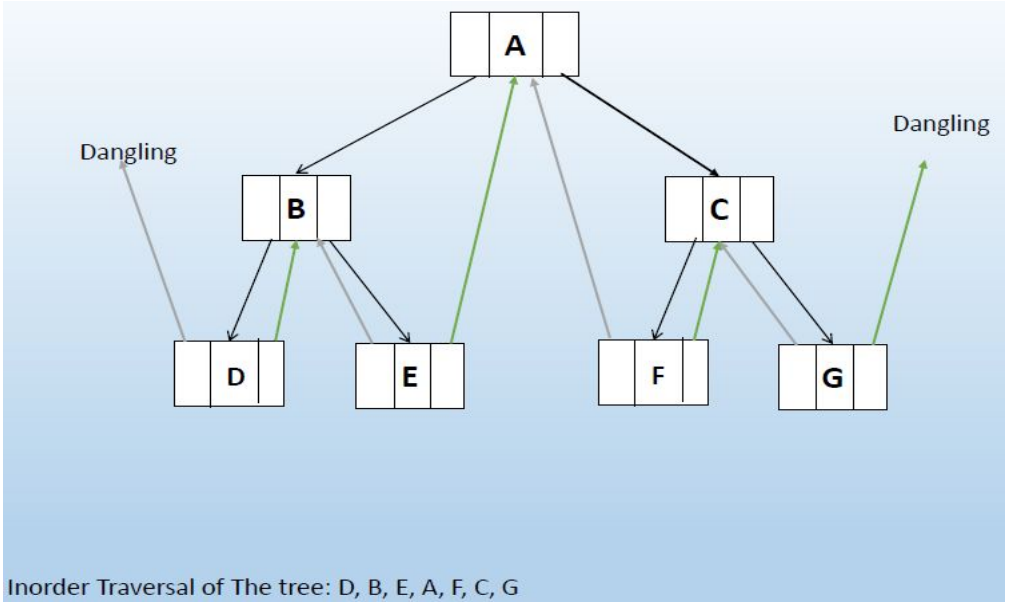
Single Threaded Binary Tree: One-Way

- We will use the right thread only in this case.
- To implement threads we need to use in-order successor of the tree.



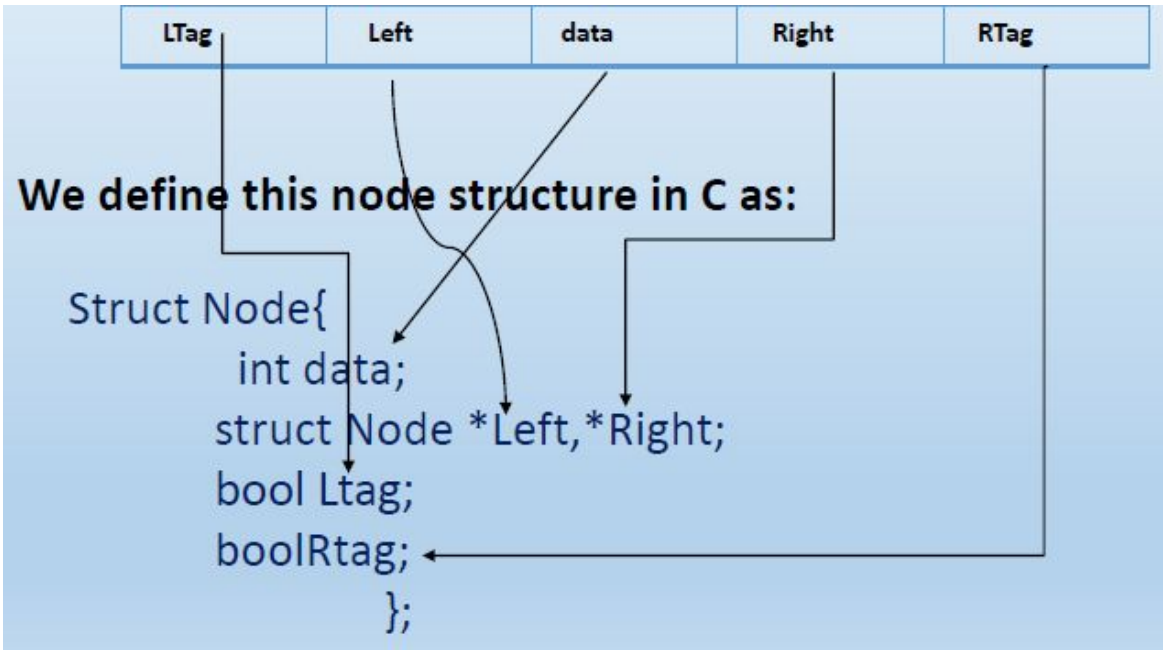
Double Threaded Binary Tree: Two-Way

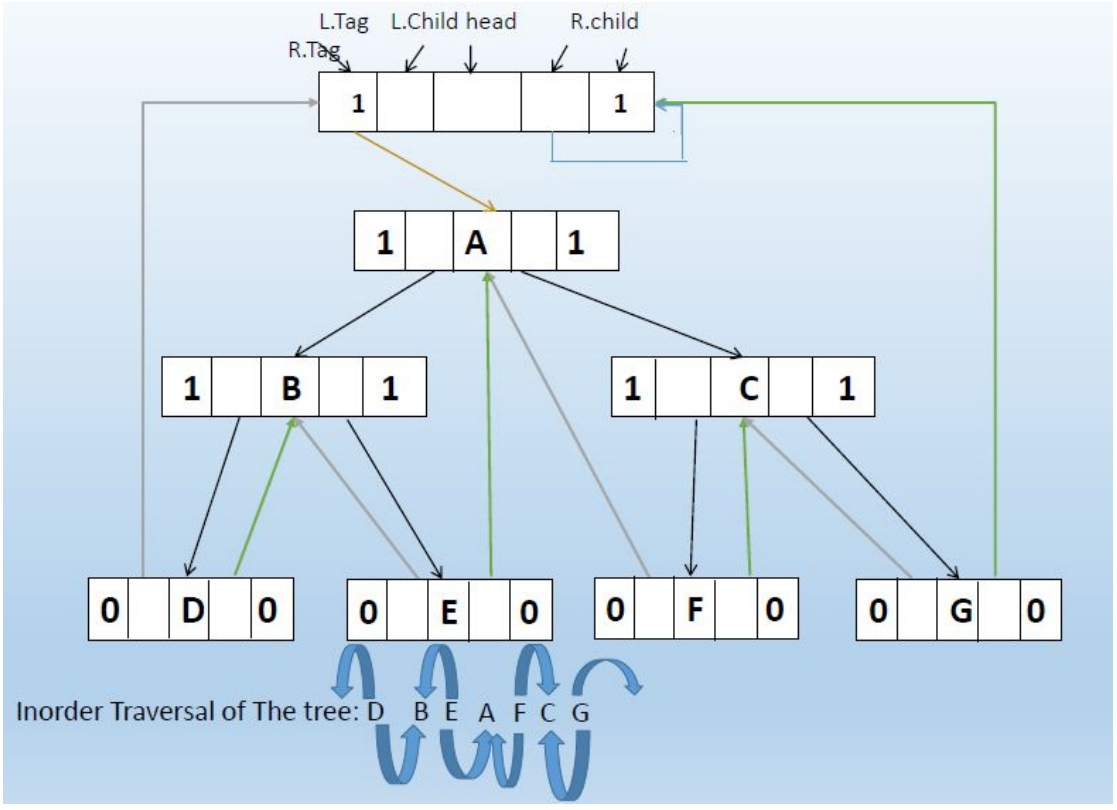
Two-way threading has left pointer of the first node and right pointer of the last node will contain the null value. The header nodes is called two-way threading with header node threaded binary tree.



Structure of Thread Binary Tree - Node structure

Assume each node has five fields:

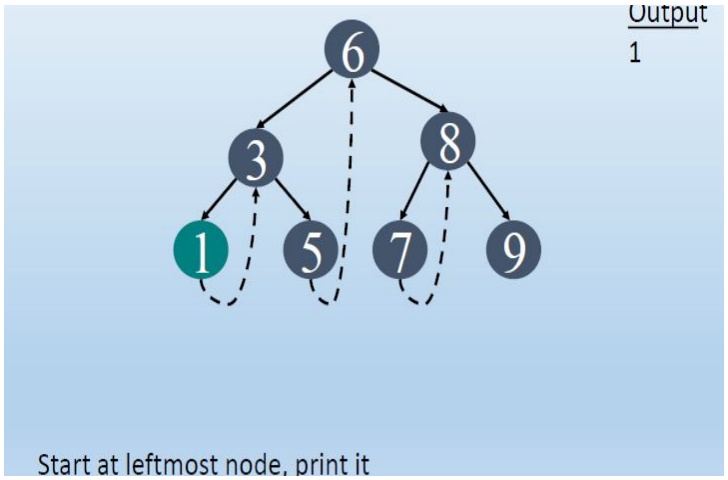


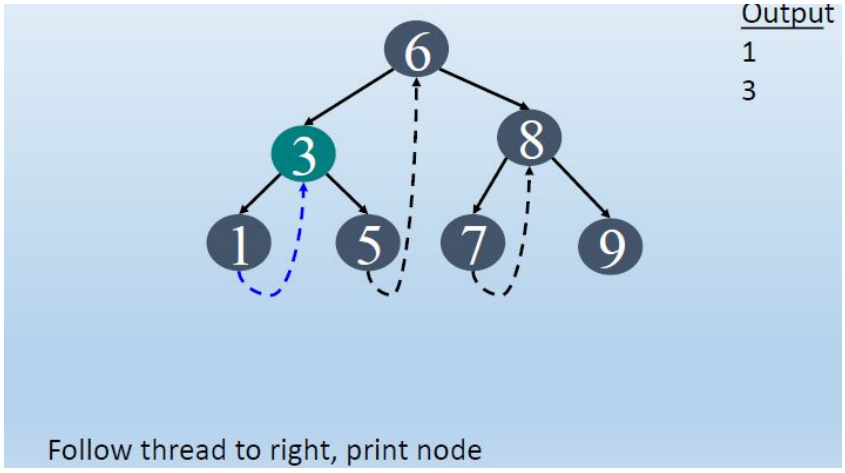


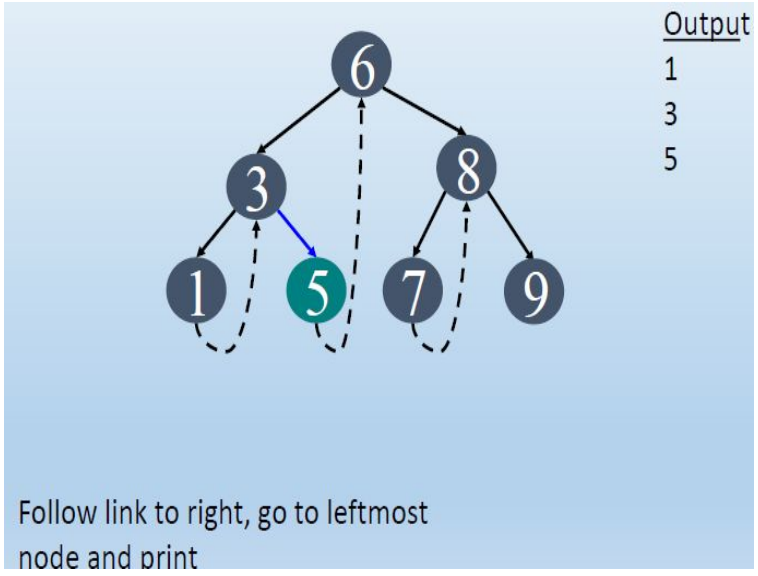
Threaded Tree Traversal

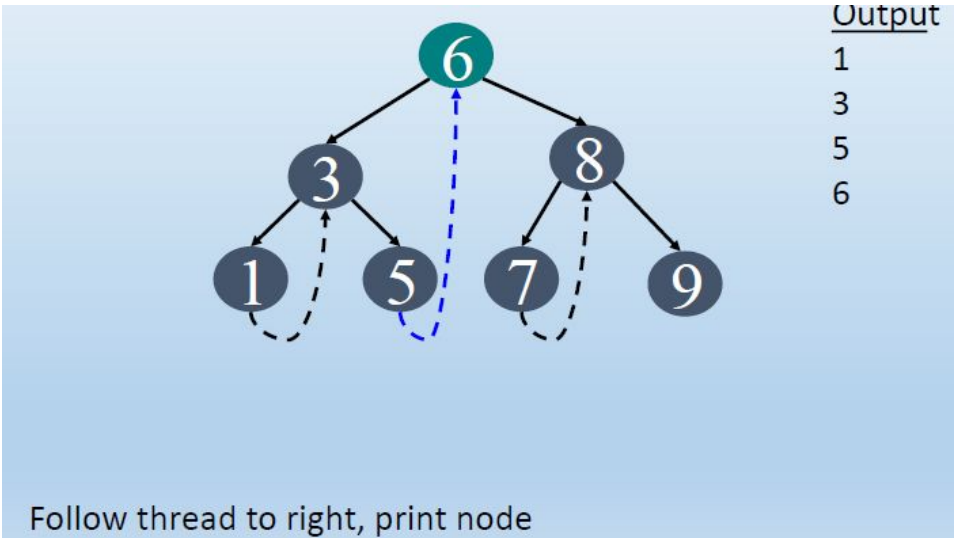
We start at the leftmost node in the tree, print it, and follow its right thread

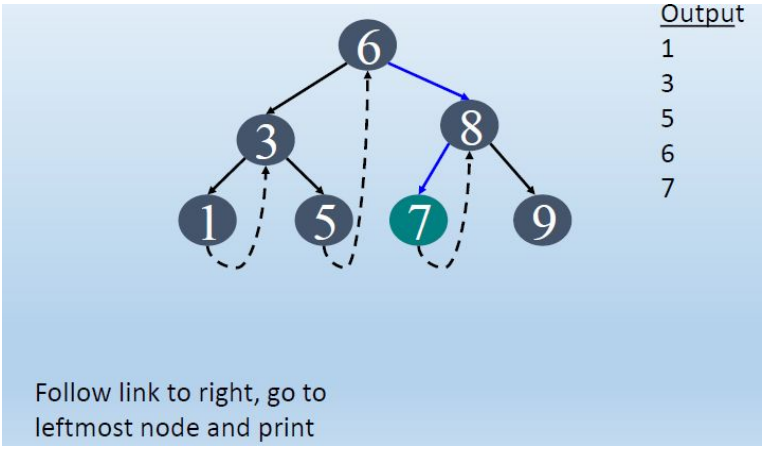
- If we follow a thread to the right, we output the node and continue to its right.
- If we follow a link to the right, we go to the leftmost node, print it, and continue.

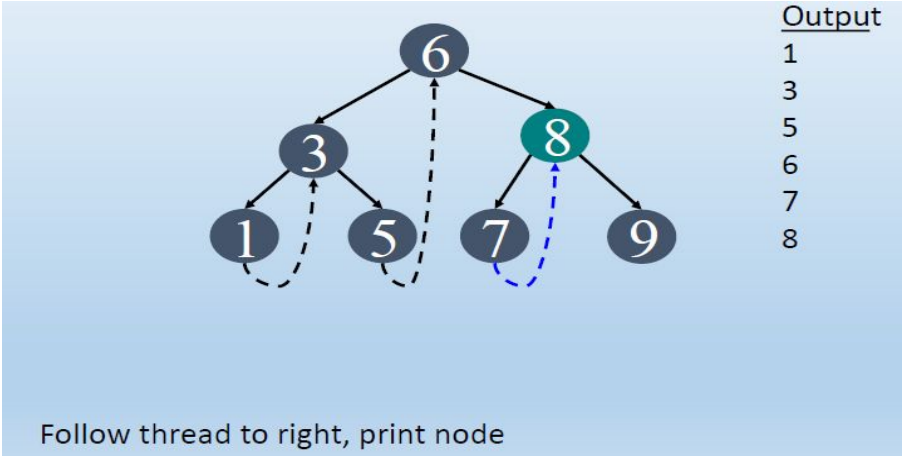

Output
1
6
3
8
1
5
7
9
Start at leftmost node, print it

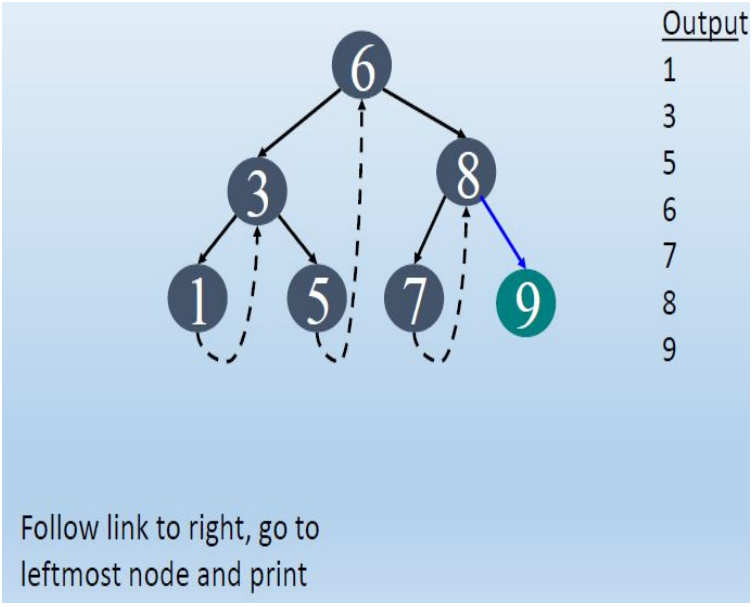

Output
1
3
6
3
8
1
5
7
9
Follow thread to right, print node


Output
1
3
5
6
3
8
1
5
7
9
Follow link to right, go to leftmost node and print

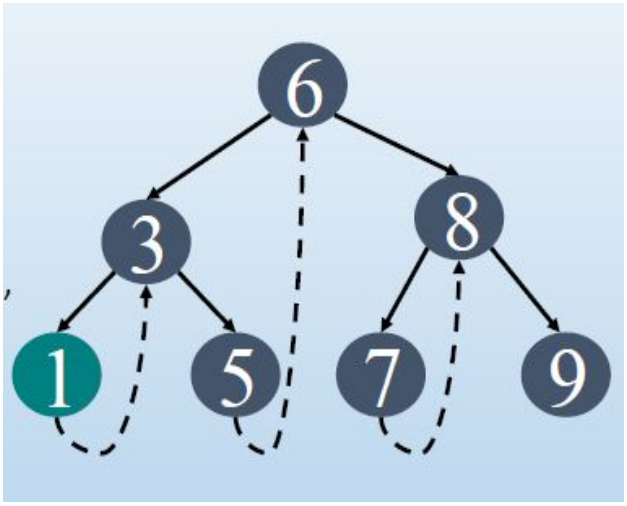

6
3
8
1
5
7
9
Output
1
3
5
6
Follow thread to right, print node


6
3
8
1
5
7
9
Output
1
3
5
6
7
Follow link to right, go to leftmost node and print


6
3
8
1
5
7
9
Output
1
3
5
6
7
8
Follow thread to right, print node




```
void inOrder(struct Node *root)
{
struct Node *cur = leftmost(root);
while (cur != NULL)
{
printf("%d ", cur->data);
if (cur->rightThread)      // If this node is a thread node,then go to inorder successor
cur = cur->right;
else               // Else go to the leftmost child
in right subtree
cur = leftmost(cur->right);
}
}
```



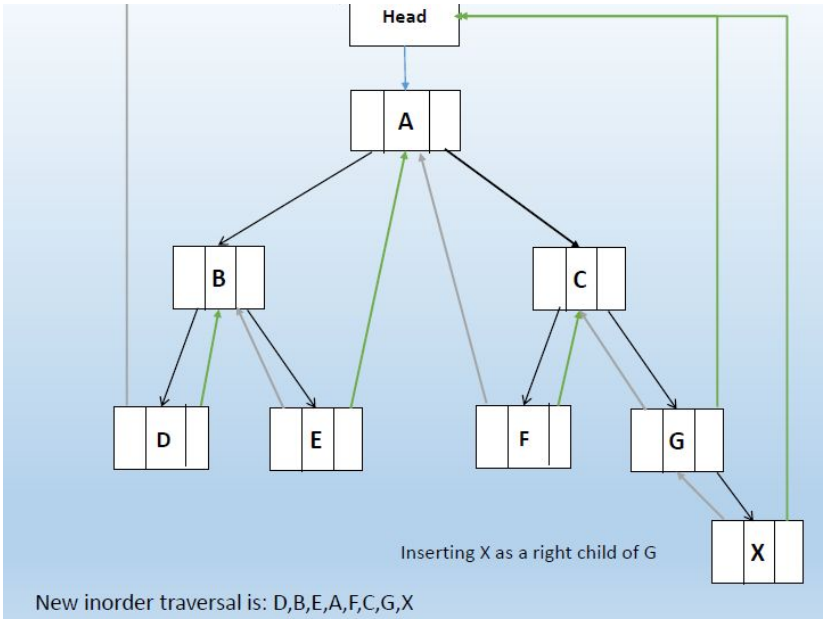
Threaded Binary Trees	**Normal Binary Trees**
In threaded binary trees, The null pointers are used as thread	In a normal binary trees, the null pointers remains null.
We can use the null pointers which is a efficient way to use computers memory.	We can't use null pointers so it is a wastage of memory.
Traversal is easy. Completed without using stack or recursive function.	Traverse is not easy and not memory efficient.
Structure is complex.	Less complex than Threaded binary tree.
Insertion and deletion takes more time.	Less Time consuming than Threaded Binary tree.

Inserting a node to Threaded Binary Tree:

Inserting a node X as the right child of a nodes.

1st Case:

- If G has an empty right subtree, then the insertion is simple



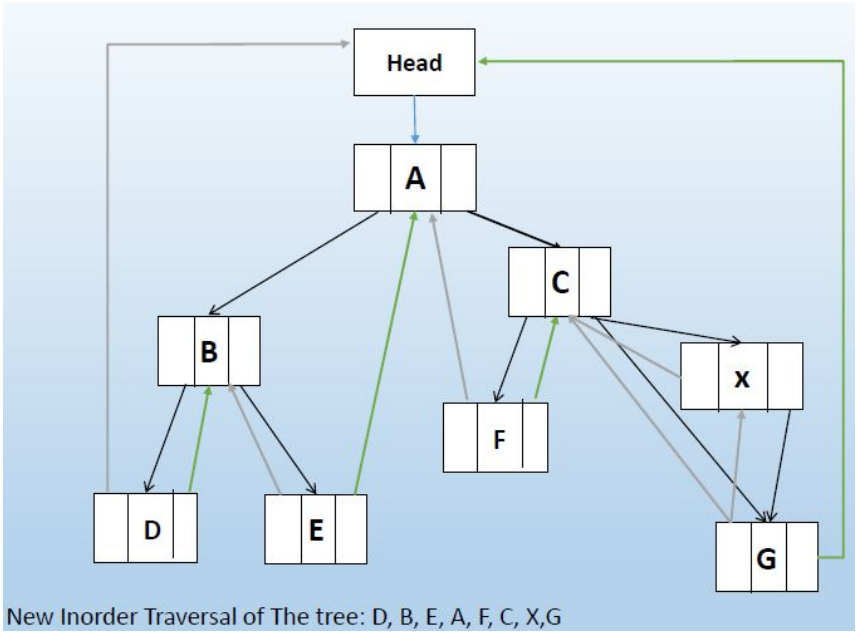
Inserting a node to Threaded Binary Tree:

Inserting a node X as the right child of a nodes.

2nd Case:

If the right subtree of C is not empty, then this right child is made the right child of X after

insertion



Advantages

1. It helps to get linear traversal of elements in the tree.
2. Linear traversal using Threaded Binary Tree eliminates the usage of Stacks concept which in turn saves a lot of memory.
3. Any node can easily identify it's parent as it has got a thread no need to use explicit parent pointers.
4. The nodes in a threaded binary tree have links to in-order predecessor and successor the traversal in both forward and backward direction can be achieved easily.

Disadvantage

1. This makes the Tree more complex .
2. More prone to errors when both the child are not present & both values of nodes pointer to their ancestors.
3. Lot of time consumes when deletion or insertion is performed.