Threaded Binary Tree

- Presented to:
- Md. Shamsujjoha
  - *Senior Lecturer of CSE department*
  - *East West University*

- Presented by:
  - Md. Khabbab Hossain Tusher

- *Date:03/04/2016*

Threaded binary tree

OVERVIEW:

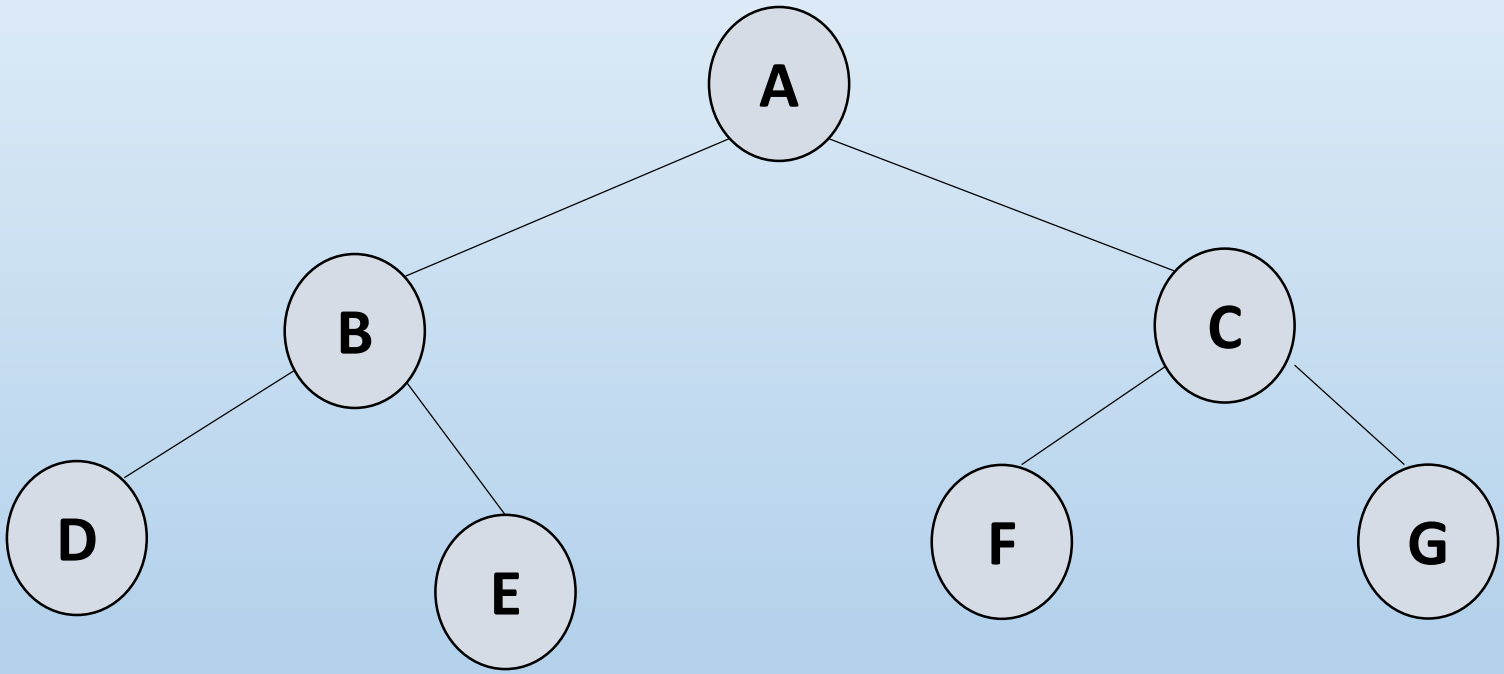
- Introduction
  - Definition
  - Example of Threaded BT.
- Types & Structure
  - One-way .
  - Double-way.
  - Structure.
- Traversal
  - Algorithm for Traversal
  - Traversal Example
- Inserting
  - Algorithm for Inserting
  - Inserting Example

- Comparison With Binary Tree
- Advantages and Disadvantages
- Why Threaded BT are used?
- Conclusion
- Reference

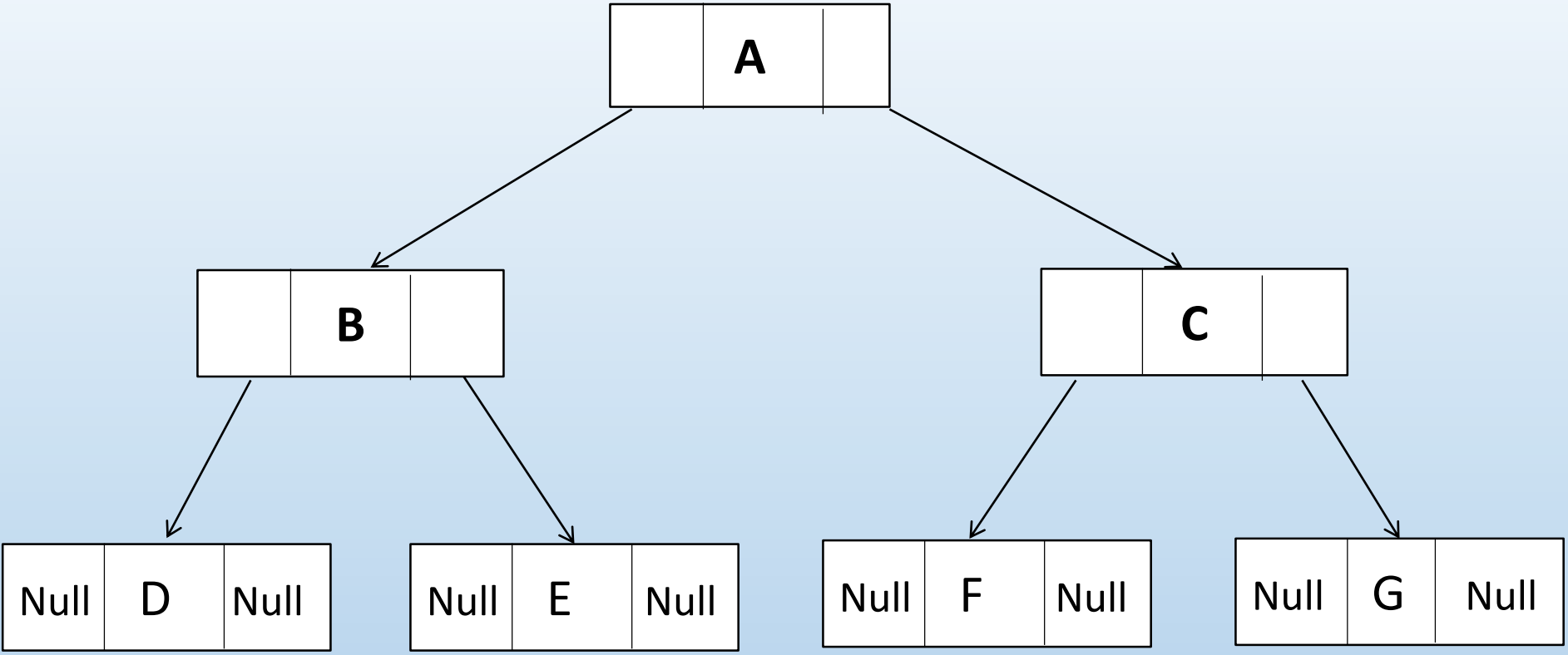
Definition

A binary search tree in which each node uses an otherwise-empty left child link to refer to the node's in-order predecessor and an empty right child link to refer to its in-Order Successor.

A Simple Binary Tree



Threaded binary tree



Threaded Binary Tree

- In above 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.
- A. J. perils & C. Thornton jointly 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.

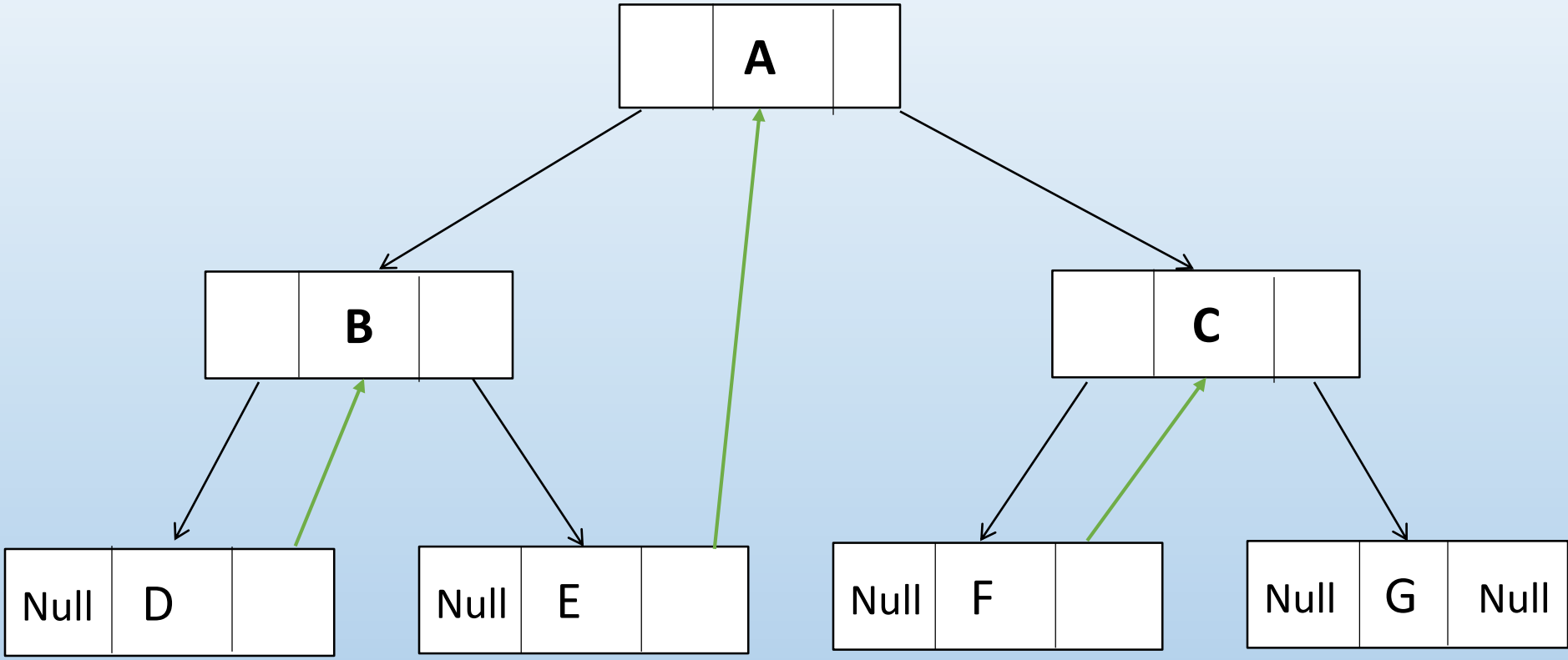
- And binary tree with such pointers are called threaded tree.
- In the memory representation of a threaded binary tree, it is necessary to distinguish between a normal pointer and a thread.

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

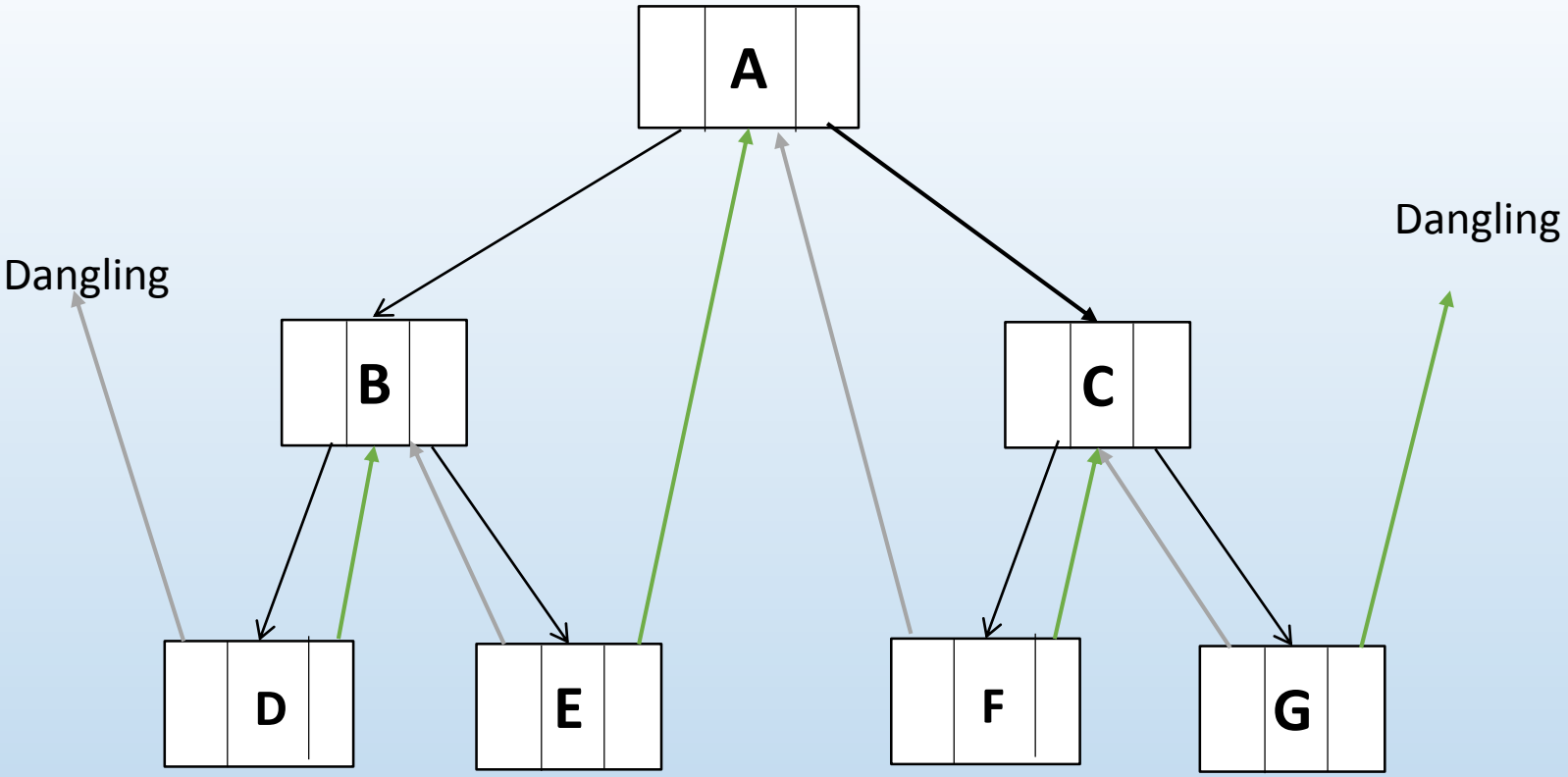
Threaded Binary Tree: One-Way



Inorder Traversal of The tree: D, B, E, A, F, C, G

Two way Threaded Tree/Double Threads

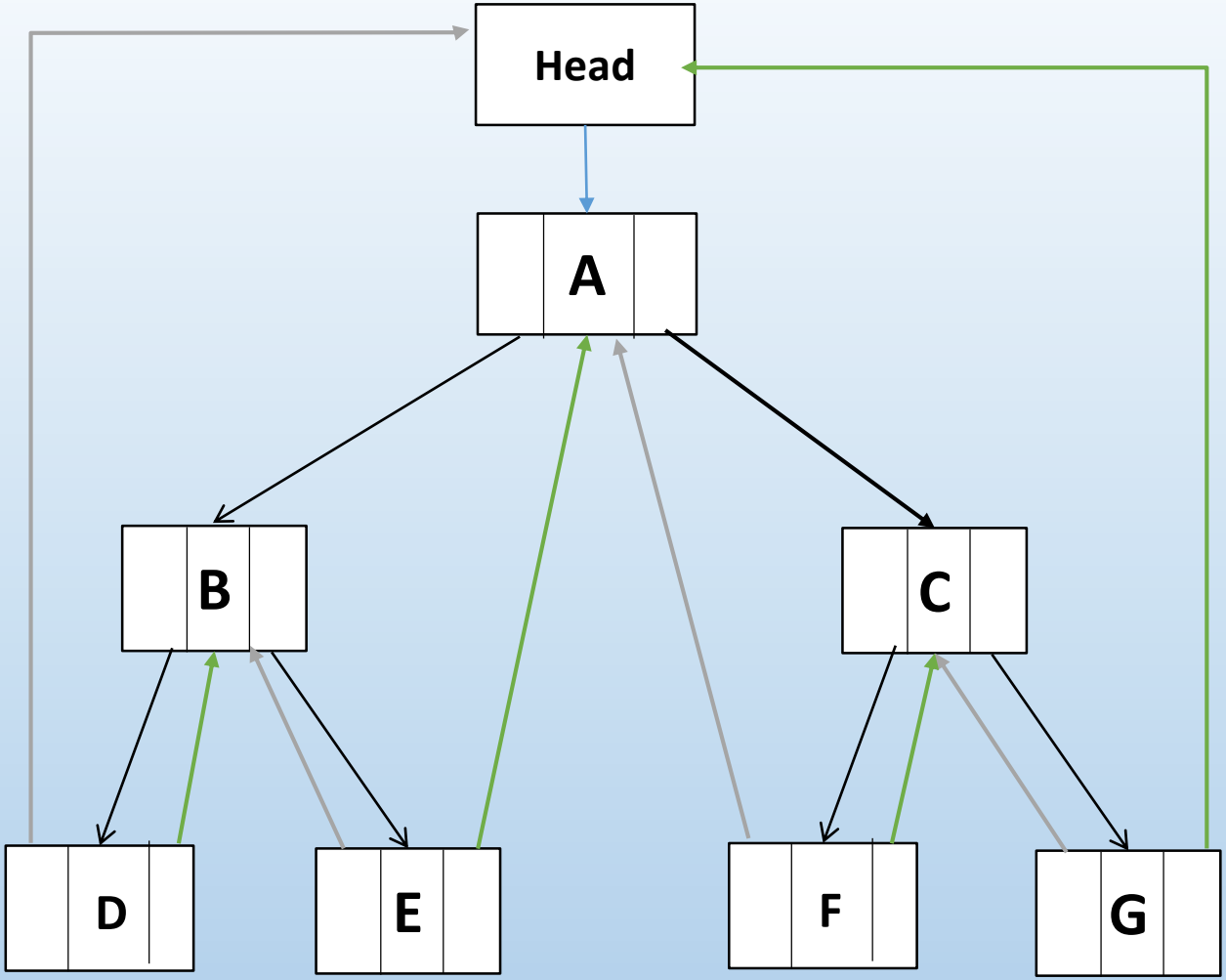
- Again two-way threading has left pointer of the first node and right pointer of the last node will contain the null value. The header nodes is called two-way threading with header node threaded binary tree.



Inorder Traversal of The tree: D, B, E, A, F, C, G

- Dangling can be solved as follows

- Introduce a header node.
- The left and right pointer of the header node are treated as normal links and are initialized to point to header node itself.

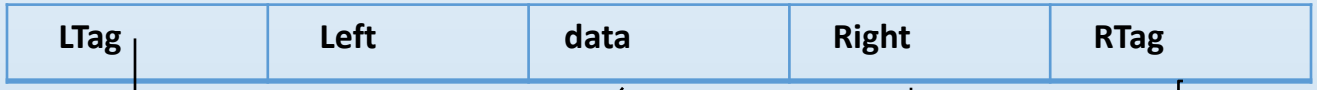


Inorder Traversal of The tree: D, B, E, A, F, C, G

Structure of Thread BT

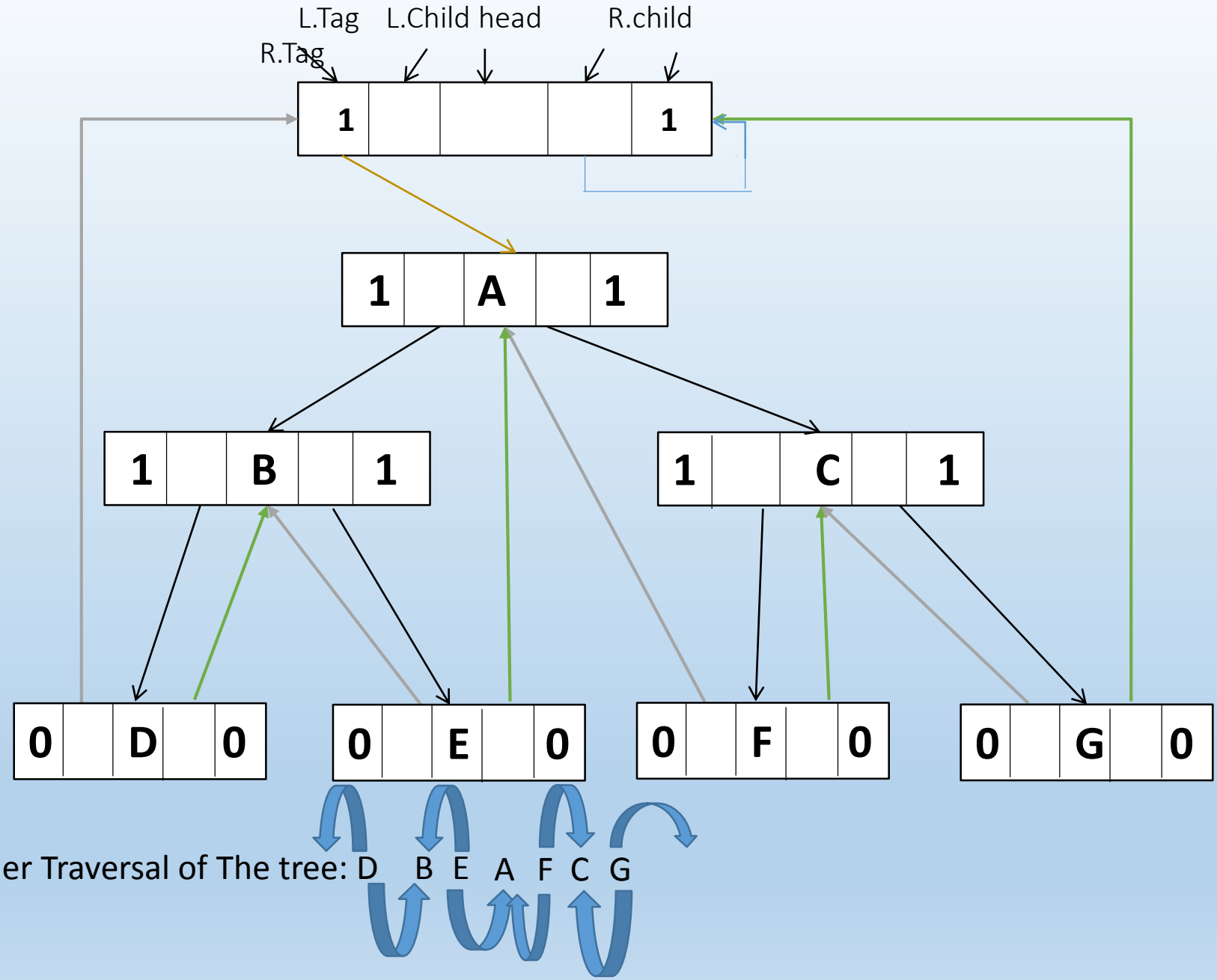
Node structure

For the purpose of our evaluation algorithm, we assume each node has five fields:



We define this node structure in C as:

```
Struct Node{
        int data;
    struct Node *Left,*Right;
    bool Ltag;
    boolRtag;
            };
```

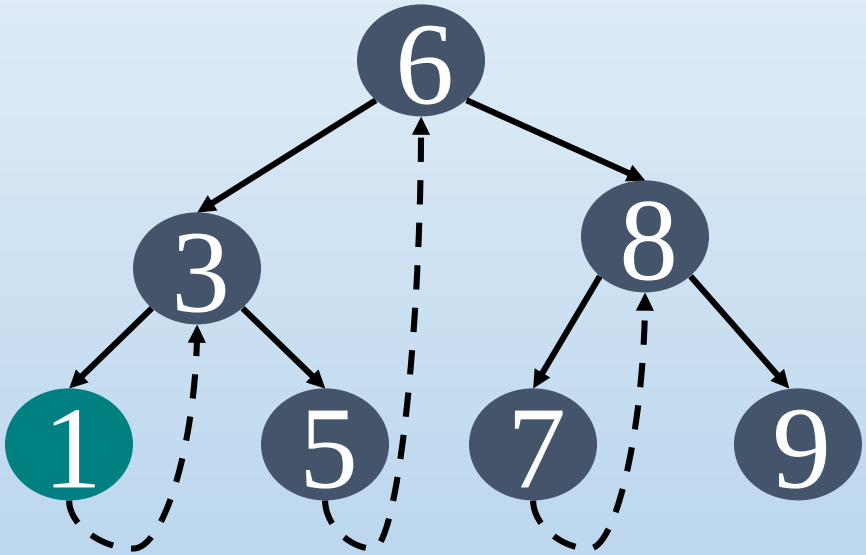


Inorder Traversal of The tree: D B E A F C G

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.

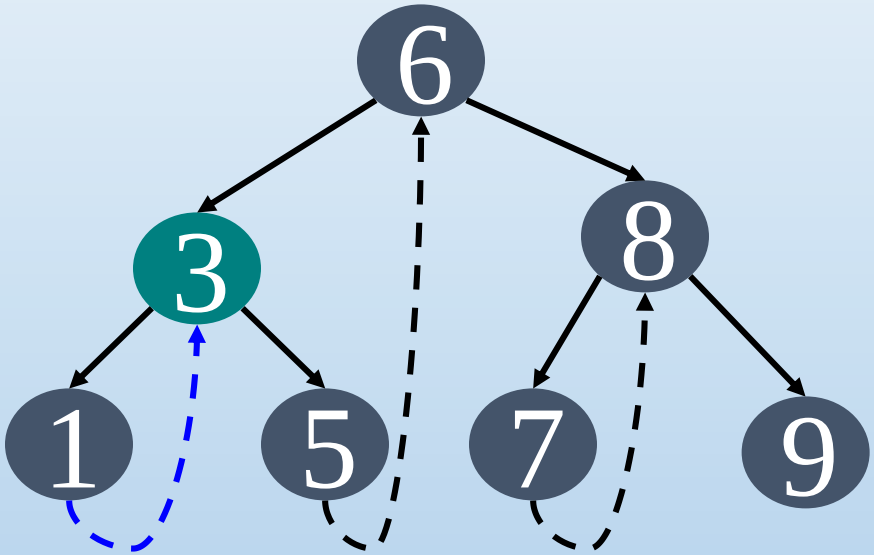
THREADED TREE TRAVERSAL



Output
1

Start at leftmost node, print it

THREADED TREE TRAVERSAL



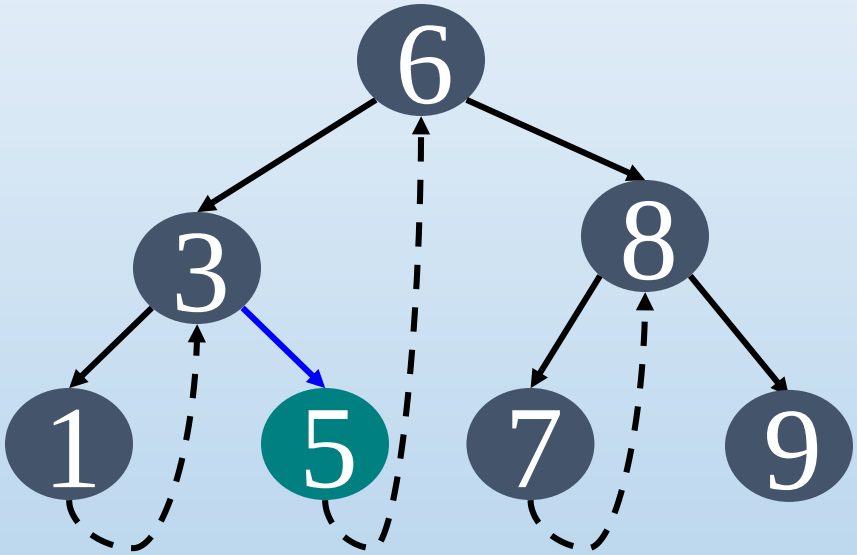
Output

1

3

Follow thread to right, print node

THREADED TREE TRAVERSAL



Output

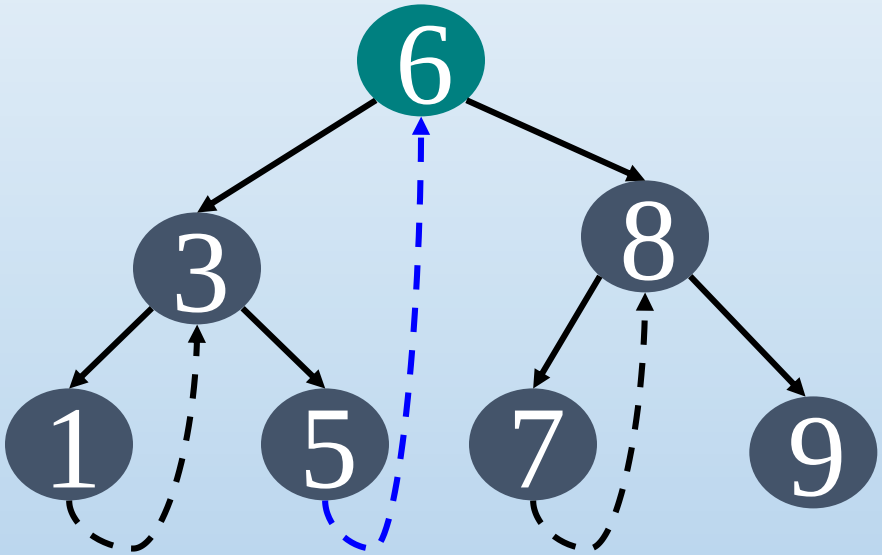
1

3

5

Follow link to right, go to leftmost node and print

THREADED TREE TRAVERSAL



Output

1

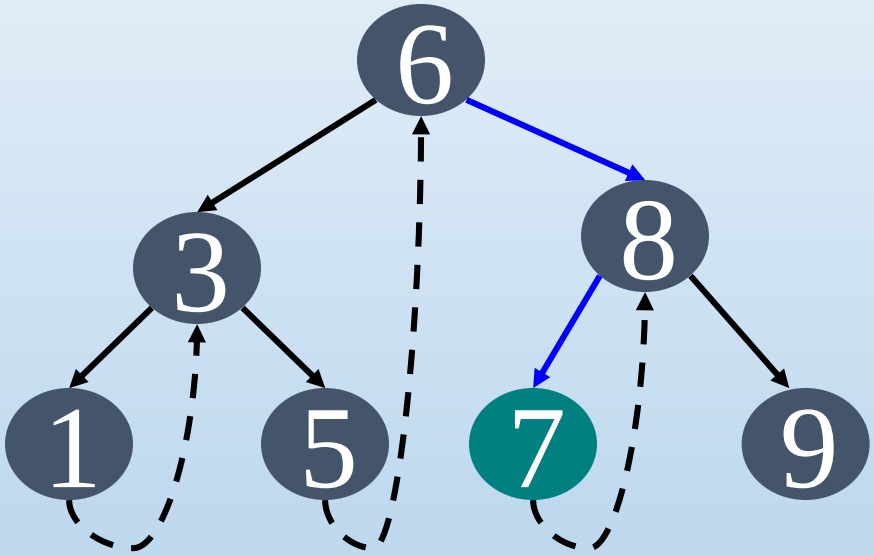
3

5

6

Follow thread to right, print node

THREADED TREE TRAVERSAL



Output

1

3

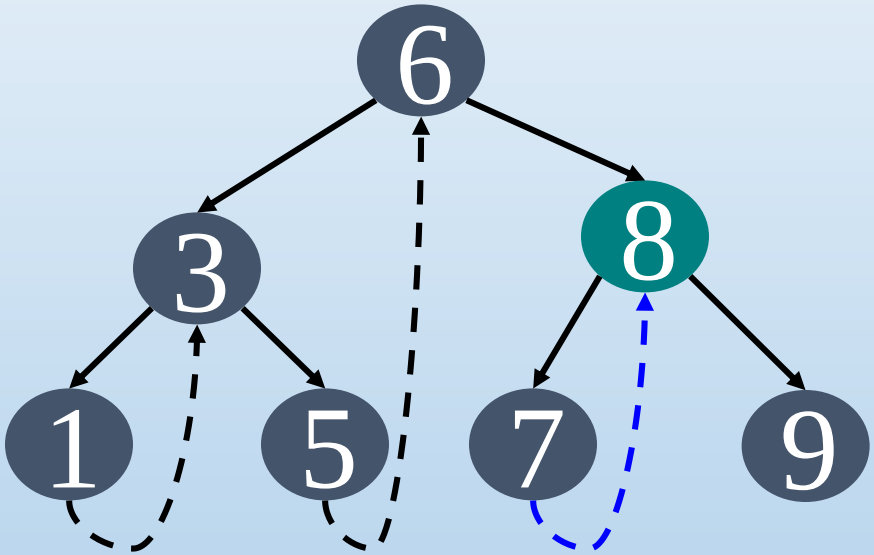
5

6

7

Follow link to right, go to leftmost node and print

THREADED TREE TRAVERSAL

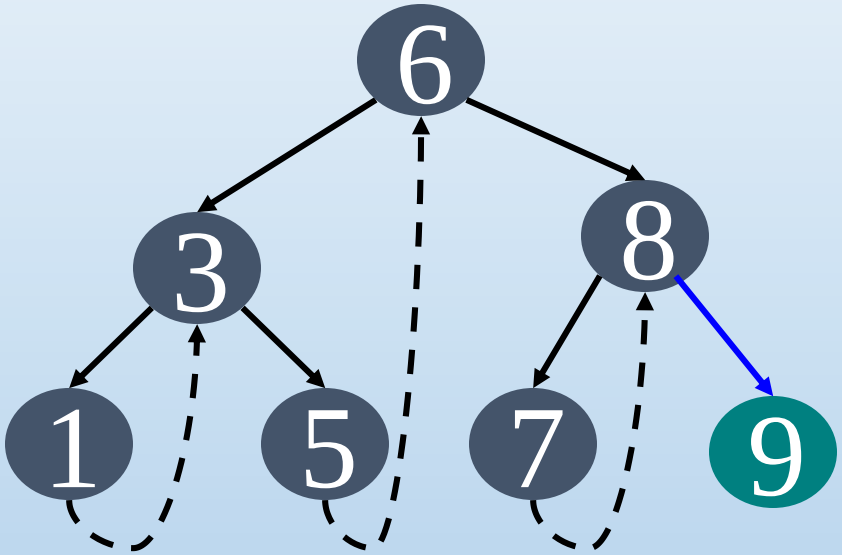


Output

1
3
5
6
7
8

Follow thread to right, print node

THREADED TREE TRAVERSAL



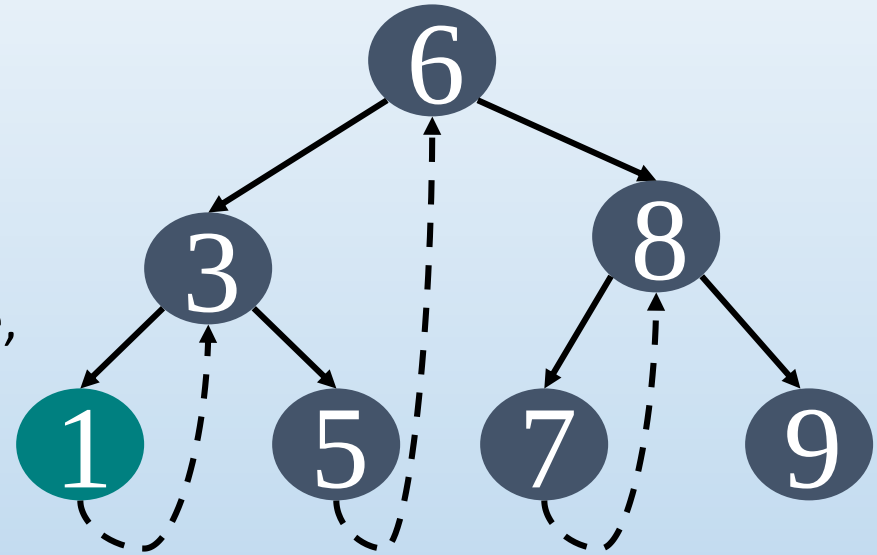
Output

1
3
5
6
7
8
9

Follow link to right, go to leftmost node and print

```
void inOrder(struct Node *root)
{
   struct Node *cur = leftmost(root);
   while (cur != NULL)
   {
       printf("%d ", cur->data);

       // If this node is a thread node, then go to
       // inorder successor
       if (cur->rightThread)
           cur = cur->right;
       else // Else go to the leftmost child in right subtree
           cur = leftmost(cur->right);
   }
}
```



Comparison of Threaded BT

Threaded Binary Trees

- In threaded binary trees, The null pointers are used as thread.
- We can use the null pointers which is a efficient way to use computers memory.
- Traversal is easy. Completed without using stack or reccursive function.
- Structure is complex.
- Insertion and deletion takes more time.

Normal Binary Trees

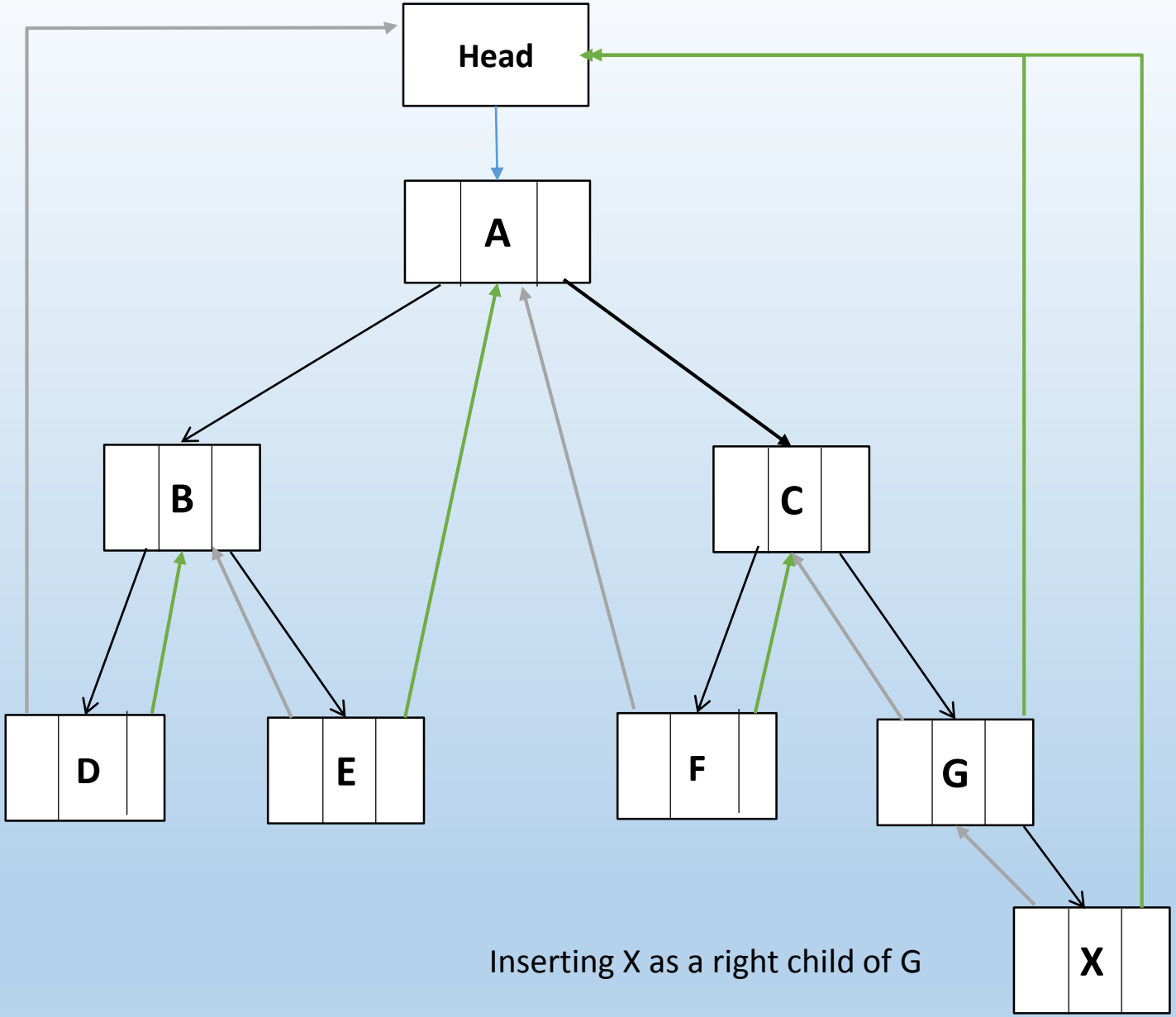
- In a normal binary trees, the null pointers remains null.
- We can't use null pointers so it is a wastage of memory.
- Traverse is not easy and not memory efficient.
- Less complex than Threaded binary tree.
- Less Time consuming than Threaded Binary tree.

- 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

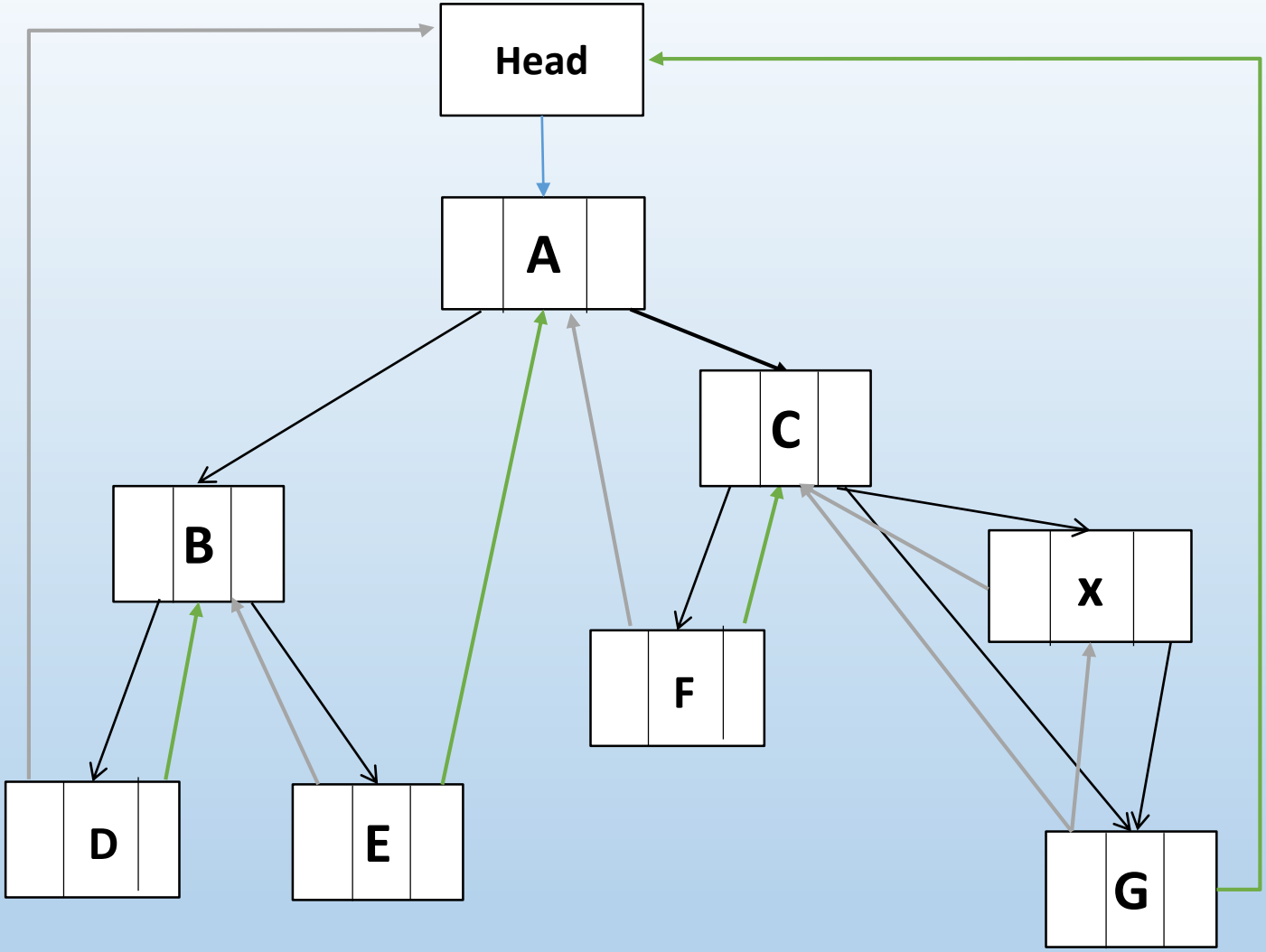


Inserting X as a right child of G

New inorder traversal is: D,B,E,A,F,C,G,X

2nd Case:

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



New Inorder Traversal of The tree: D, B, E, A, F, C, X,G

Threaded binary tree

Advantage

- 1. By doing threading we avoid the recursive method of traversing a Tree , which doesn't use of stack and consumes a lot of memory and time .
- 2 . The node can keep record of its root .
- 3. Backward Traverse is possible.
- 4. Applicable in most types of the binary tree.

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.

APPLICATIONS

- Same as any kind of Binary Tree.
- Used in search and Traverse based work

Conclusion

- Excellent concept in modern computer science.
- Saves Time and memory.
- Traversal and search are very efficient.
- Insertion and deletion are not so efficient.

References

- http://www.geeksforgeeks.org/inorder-tree-traversal-without-recursion-and-without-stack/
- http://www.delorie.com/gnu/docs/avl/libavl_183.html
- http://www.math-cs.gordon.edu/courses/cs321/lectures/threaded.html
- https://prezi.com/1yitau0wnwlg/threaded-binary-tree-in-data-structures/
- http://stackoverflow.com/questions/6744770/confusion-with-threaded-binary-tree
- Various Text Book for data structure.

THANK YOU

ANY QUESTION OR SUGGESTION?

Contacts us
:tkhabbab@gmail.com/
khabbabcse@ewu.edu.bd