Threads

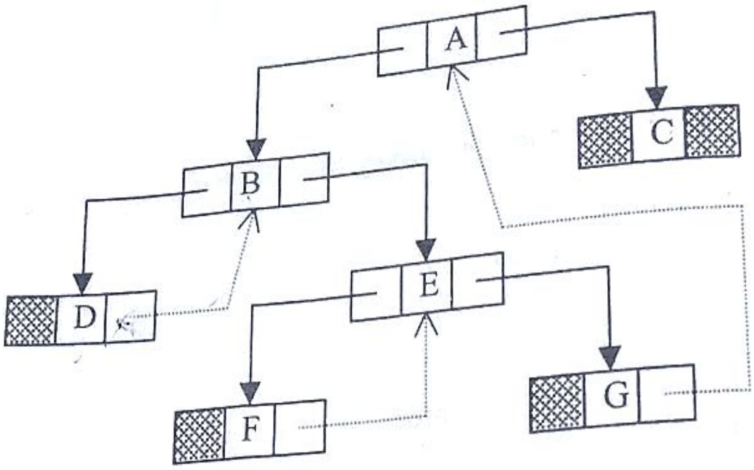
In linked representation of binary tree we can see that most of the nodes have NULL value in their left and right pointer fields. It will be useful to use these pointer fields to keep some other information for operations in binary tree. Traversing is the most common operation in binary tree. We can use these pointer fields to contain the address pointer which points to the nodes higher in the tree. Such pointer which keeps the address of the nodes higher in tree is called thread. A binary tree which implements these pointers is called threaded binary tree.

We can have threading corresponding to any of the three traversals i.e postorder, preorder and inorder. There may be two types of inorder threading, one-way inorder threading and two-way inorder threading. In one-way inorder threading, right field of the node will keep the thread pointer which will point to the next node in the sequence of the inorder traversal or we can say that right thread will point to the inorder successor of the node.

In two-way inorder threading left field of node will also keep the thread pointer which will point to the previous node in the sequence of inorder traversal or we can say left thread will point to the inorder predecessor of the node.

If we use right field of node to take the thread then this is called right in-threaded binary tree. When we use left field of node to take the thread then this is called left in-threaded binary tree. If both left and right fields are used for threading then it is called fully threaded binary tree or in-threaded binary tree.

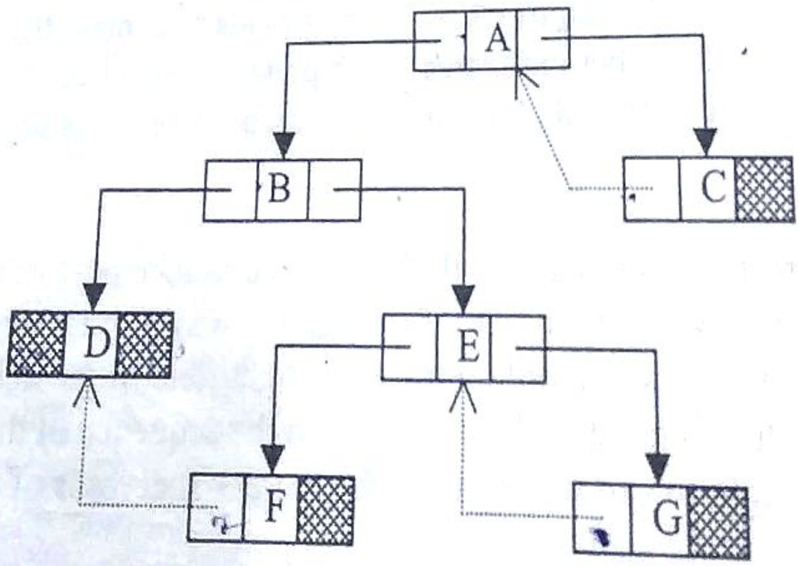
Similarly, we can have right pre-threaded and left pre-threaded tree corresponding to preorder traversal.



Right-in –threaded binary tree

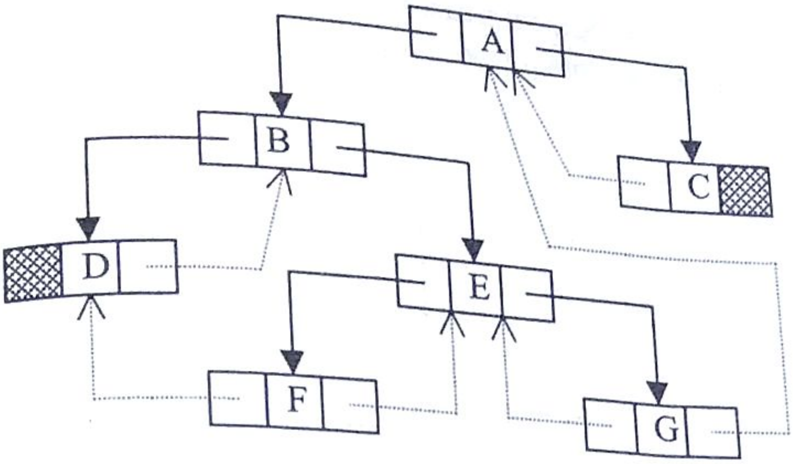
Inorder Traversal D B F E G A C

In this tree right child of node has threads which point to the next node in the sequence of the inorder traversal. For example here thread of D points to B which is inorder successor of D in sequence of inorder traversal.



Left-in-threaded binary tree

In this tree left child of node has threads which point to the previous node in the seque of the inorder traversal. For example here thread of F points to B which is inorder predecessor of F in inorder traversal



Fully in-threaded binary tree

Here left thread of F points to B which is inorder predecessor of F and right thread of F points to E which is inorder successor of F.

The structure of a node in a two way in threaded binary tree will be as

```
typedef enum {thread, link } boolean;
struct node
{
        struct node *left_ptr ;
        boolean left;
        int info;
        boolean node *right_ptr ;
        boolean right;

};
```

Here we have taken two boolean members left and right to differentiate between a thread and link. These members can take values thread or link.

left =link	pointer left_ptr points to the left child of the node.
left =thread	pointer left_ptr is a thread pointing to inorder predecessor of the node.
right =link	pointer right_ptr points to the right child of the node.
right =thread	pointer right_ptr is a thread pointing to inorder successor of the node.

We have seen that if left or right pointer of a node is NULL then it is made a thread pointing to the inorder predecessor or inorder successor of the node. First node in inorder traversal has no predecessor and last node has no successor. So the left pointer of the leftmost node which is the first node in inorder traversal and the right pointer of the rightmost node which is the last node in inorder traversal contains NULL. So we can take a dummy node called the header node and we will represent our tree as the left subtree of this header node. Left pointer of this header node will point to the root node of our tree. When our tree will be empty then left pointer of this header node will be a thread pointing to itself.