TRAVERSING A BINARY TREE

Traversing a binary tree is the process of visiting each node in the tree exactly once in a systematic way. Unlike linear data structures in which the elements are traversed sequentially, tree is a non-linear data structure in which the elements can be traversed in many different ways.

Pre-order Traversal

To traverse a non-empty binary tree in pre-order, the following operations are performed recursively at each node. The algorithm works by:

1. Visiting the root node,
2. Traversing the left sub-tree, and finally
3. Traversing the right sub-tree.

```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:            Write TREE -> DATA
Step 3:            PREORDER(TREE -> LEFT)
Step 4:            PREORDER(TREE -> RIGHT)
       [END OF LOOP]
Step 5: END
```

NOTE: Pre-order traversal is also called as depth-first traversal or NLR traversal algorithm (Node-Left-Right).

In-order Traversal

To traverse a non-empty binary tree in in-order, the following operations are performed recursively at each node. The algorithm works by:

1. Traversing the left sub-tree,
2. Visiting the root node, and finally
3. Traversing the right sub-tree.

```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:           INORDER(TREE -> LEFT)
Step 3:           Write TREE -> DATA
Step 4:           INORDER(TREE -> RIGHT)
       [END OF LOOP]
Step 5: END
```

NOTE: In-order traversal is also called as symmetric traversal (or) LNR traversal algorithm (Left-Node-Right).

Post-order Traversal

To traverse a non-empty binary tree in post-order, the following operations are performed recursively at each node.

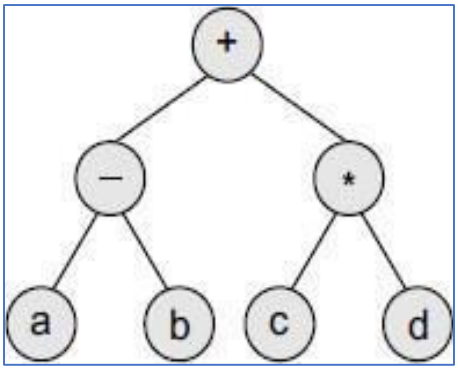
The algorithm works by:

1. Traversing the left sub-tree,
2. Traversing the right sub-tree, and finally
3. Visiting the root node.

```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:           POSTORDER(TREE -> LEFT)
Step 3:           POSTORDER(TREE -> RIGHT)
Step 4:           Write TREE -> DATA
       [END OF LOOP]
Step 5: END
```

NOTE: Post-order algorithm is also known as the LRN traversal algorithm (Left- Right- Node)

Example 1: Find the In-order, Pre-order and post-order traversal of given tree



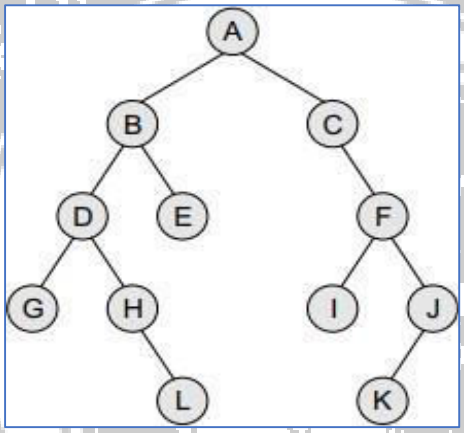
Solution

Pre-order Traversal : + – a b * c d

In-order Traversal : a-b+c*d

Post-order Traversal : ab-cd*+

Example 2: Find the In-order, Pre-order and post-order traversal of given tree



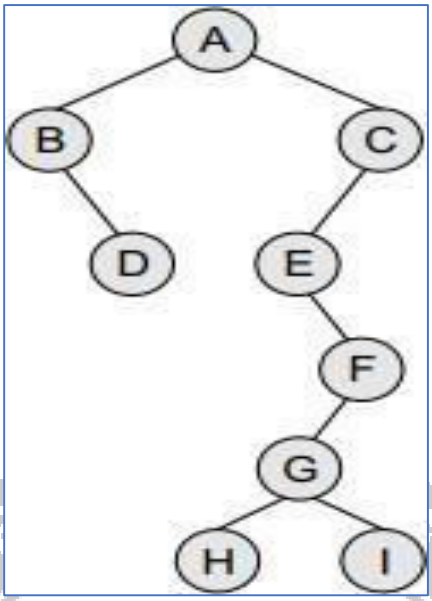
Solution

Pre-order Traversal : A, B, D, G, H, L, E, C, F, I, J, and K

In-order Traversal : G, D, H, L, B, E, A, C, I, F, K, and J

Post-order Traversal : G, L, H, D, E, B, I, K, J, F, C, and A

Example 3: Find the In-order, Pre-order and post-order traversal of given tree



Solution

Pre-order Traversal : A, B, D, C, D, E, F, G, H, and I

In-order Traversal : B, D, A, E, H, G, I, F, and C

Post- order Traversal : D, B, H, I, G, F, E, C, and A

Level-order Traversal

In level-order traversal, all the nodes at a level are accessed before going to the next level. This algorithm is also called as the breadth-first traversal algorithm. Consider the trees given in Fig. and note the level order of these trees.