Traversal in Binary Tree:

Traversal is the process to visit all the nodes of a tree at least once.

Generally, we traverse a tree to search or locate a given item or key in the tree or to print all the values it contains.

There are three ways used to traverse a tree –

A. Pre-order Traversal
B. In-order Traversal
C. Post-order Traversal

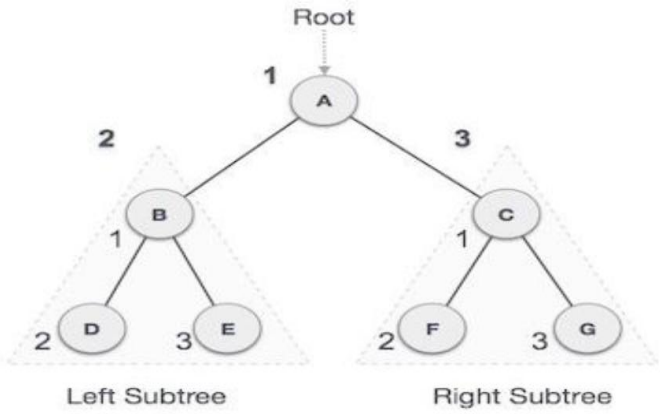
A. Pre-order Traversal:

In this traversal method, the root node is visited first, then the left subtree and finally the right subtree using the pre-order traversal.

Steps:

1. Visit root
2. Traverse left sub tree in pre-order
3. Traverse right sub tree in pre-order

Example:



Output: The output of pre-order traversal for above tree will be −

A B D E C F G

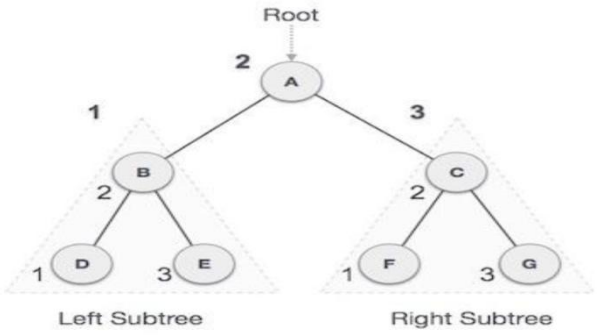
B. **In-order Traversal:**

In this traversal method, the left subtree is visited first, then the root and later the right sub-tree is processed using the in-order traversal.

Steps:

1. Traverse left sub tree in in-order
2. Visit root
3. Traverse right sub tree in in-order

Example:



Output: The output of in-order traversal of the above tree will be −

D B E A F C G

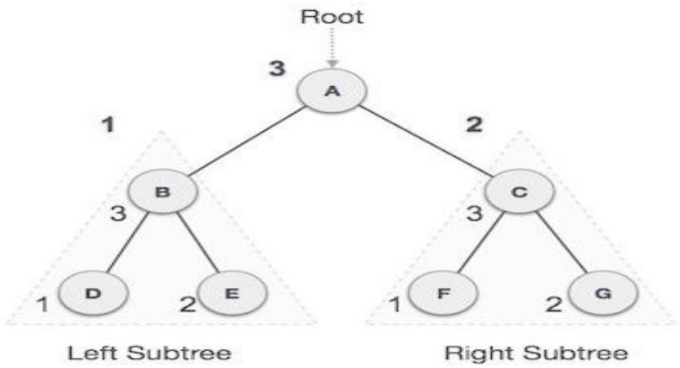
C. **Post-order traversal:**

In this traversal method, first the left subtree is traversed, then the right subtree and finally the root node using post order traversal.

Steps:

1. Traverse left sub tree in post-order
2. Traverse right sub tree in post-order
3. Visit root

Example:

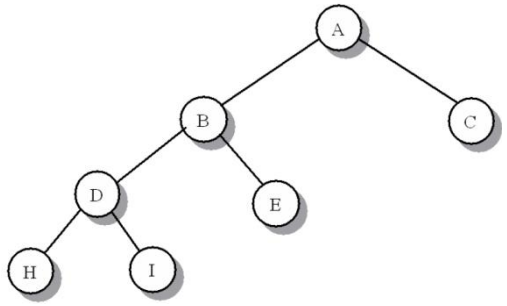


Output: The output of post order traversal of above tree will be –

D E B F G C A

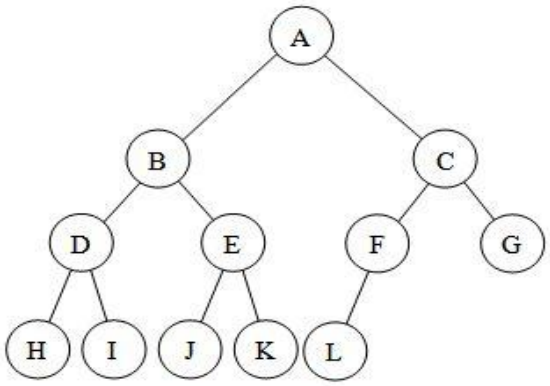
Examples:

Example-1



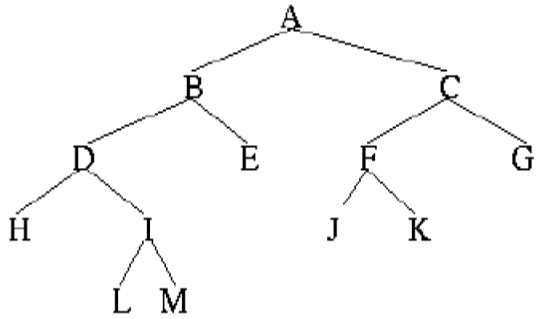
Pre-order: A, B, D, H, I, E, C
In-order: H, D, I, B, E, A, C
Post-order: H, I, D, E, B, C, A

Example-2



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

Example-3:



Pre-order: A, B, D, H, I, L, M, E, C, F, J, K, G
In-order: H, D, L, I, M, B, E, A, J, F, K, C, G
Post-order: H, L, M, I, D, E, B, J, K, F, G, C, A