

Binary Tree Traversal

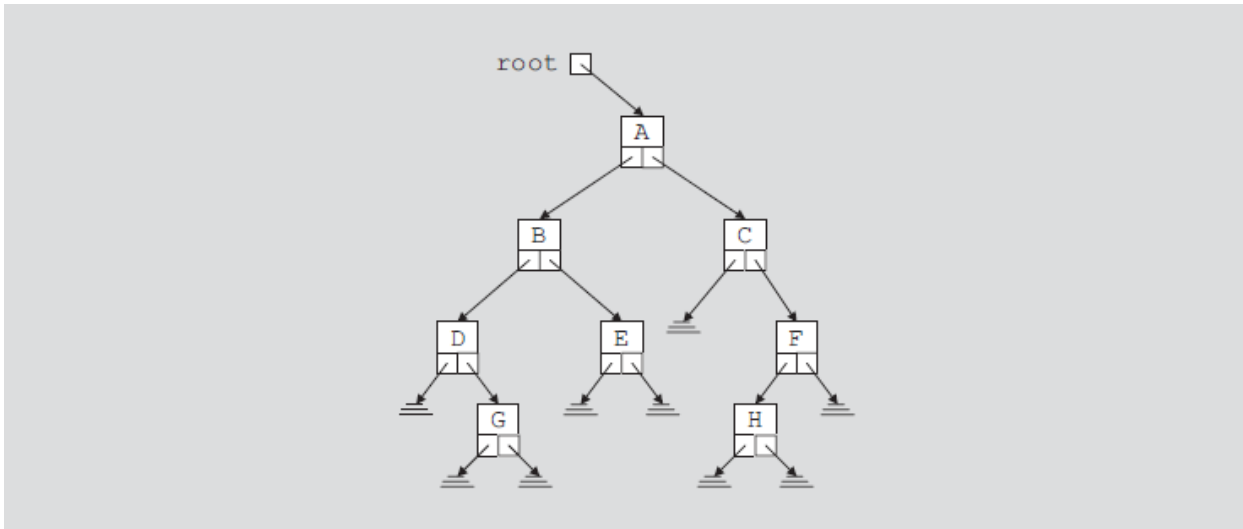


FIGURE 11-4 Binary tree

The item insertion, deletion, and lookup operations require that the binary tree be traversed. Thus, the most common operation performed on a binary tree is to traverse the binary tree, or visit each node of the binary tree. As you can see from the diagram of a binary tree, the traversal must start at the root node because there is a pointer to the root node. For each node, we have two choices:

- Visit the node first.
- Visit the sub trees first.

These choices lead to three different traversals of a binary tree—**Inorder**, **preorder**, and **postorder**.

Inorder Traversal

In an inorder traversal, the binary tree is traversed as follows:

1. Traverse the left subtree.
2. Visit the node.
3. Traverse the right subtree.

Preorder Traversal

In a preorder traversal, the binary tree is traversed as follows:

1. Visit the node.
2. Traverse the left subtree.
3. Traverse the right subtree.

Postorder Traversal

In a postorder traversal, the binary tree is traversed as follows:

1. Traverse the left subtree.
2. Traverse the right subtree.
3. Visit the node.

The listing of the nodes produced by the inorder traversal of a binary tree is called the inorder sequence. The listing of the nodes produced by the preorder traversal of a binary tree is called the preorder sequence. The listing of the nodes produced by the postorder traversal of a binary tree is called the postorder sequence.

Let us illustrate the inorder traversal of the binary tree in Figure 11-5. For simplicity, we assume that visiting a node means to output the data stored in the node.

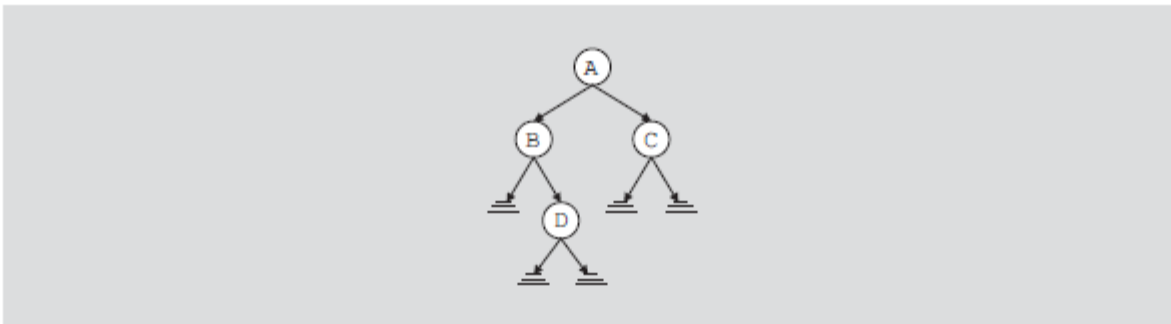


FIGURE 11-5 Binary tree for an inorder traversal

A pointer to the binary tree in Figure 11-5 is stored in the pointer variable root (which points to the node with info A). Therefore, we start the traversal at A.

1. Traverse the left subtree of A; that is, traverse $LA = \{B, D\}$.

2. Visit A.
3. Traverse the right subtree of A; that is, traverse $RA = \{C\}$.

Now, we cannot do Step 2 until we have finished Step 1.

1. Traverse the left subtree of A; that is, traverse $LA = \{B, D\}$. Now LA is a binary tree with the root node B. Because LA is a binary tree, we apply the inorder traversal criteria to LA.

- 1.1. Traverse the left subtree of B; that is, traverse $LB = \text{empty}$.
- 1.2. Visit B.
- 1.3. Traverse the right subtree of B; that is, traverse $RB = \{D\}$.

As before, first we complete Step 1.1 before going to Step 1.2.

1.1. Because the left subtree of B is empty, there is nothing to traverse.

Step 1.1 is completed, so we proceed to Step 1.2.

1.2. Visit B. That is, output B on an output device. Clearly, the first node printed is B. This completes Step 1.2, so we proceed to Step 1.3.

1.3. Traverse the right subtree of B; that is, traverse $RB = \{D\}$. Now RB is a binary tree with the root node D. Because RB is a binary tree, we apply the inorder traversal criteria to RB.

1.3.1. Traverse the left subtree of D; that is, traverse $LD = \text{empty}$.

1.3.2. Visit D.

1.3.3. Traverse the right subtree of D; that is, traverse $RD = \text{empty}$.

1.3.1. Because the left subtree of D is empty, there is nothing to traverse. Step 1.3.1 is completed, so we proceed to Step 1.3.2.

1.3.2. Visit D. That is, output D on an output device.

This completes Step 1.3.2, so we proceed to Step 1.3.3.

1.3.3. Because the right subtree of D is empty, there is nothing to traverse. Step 1.3.3 is completed.

This completes Step 1.3. Because Steps 1.1, 1.2, and 1.3 are completed, Step 1 is completed, and so we go to Step 2.

2. Visit A. That is, output A on an output device. This completes Step 2, so we proceed to Step 3.

3. Traverse the right subtree of A; that is, traverse $RA = \{C\}$. Now RA is a binary tree with the root node C. Because RA is a binary tree, we apply the inorder traversal criteria to RA.

3.1. Traverse the left subtree of C; that is, traverse $LC = \text{empty}$.

3.2. Visit C.

3.3. Traverse the right subtree of C; that is, traverse $RC = \text{empty}$.

3.1. Because the left subtree of C is empty, there is nothing to traverse. Step 3.1 is completed.

3.2. Visit C. That is, output C on an output device. This completes Step 3.2, so we proceed to Step 3.3.

3.3. Because the right subtree of C is empty, there is nothing to traverse. Step 3.3 is completed.

This completes Step 3, which in turn completes the traversal of the binary tree.

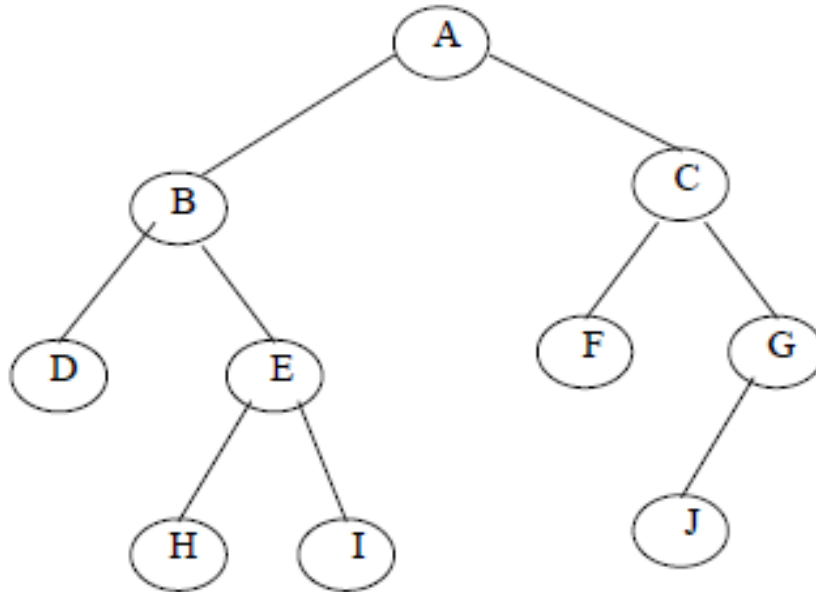
Clearly, the inorder traversal of the previous binary tree outputs the nodes in the following order:

Inorder sequence: $B D A C$

Similarly, the preorder and postorder traversals output the nodes in the following order:

Preorder sequence: $A B D C$

Postorder sequence: $D B C A$



The preorder traversal of a binary tree in Fig. is $A, B, D, E, H, I, C, F, G, J$.

The in order traversal of a binary tree in Fig. is $D, B, H, E, I, A, F, C, J, G$.

The post order traversal of a binary tree in Fig. 8.12 is $D, H, I, E, B, F, J, G, C, A$