



Tree Traversal



Binary Search Tree

- A binary tree with the property that items in the left subtree are smaller than the root and items are larger or equal in the right subtree is called a *binary search tree* (BST).
- The tree we built for searching for duplicate numbers was a binary search tree.
- BST and its variations play an important role in searching algorithms.

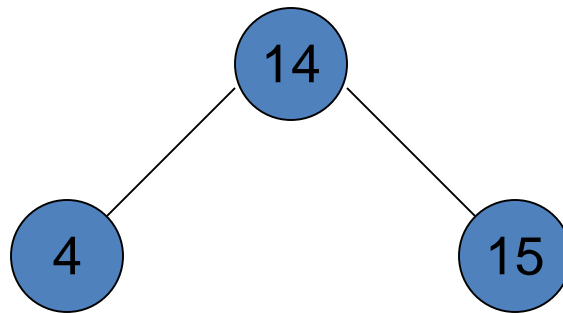


Traversing a Binary Tree

- Suppose we have a binary tree, ordered (BST) or unordered.
- We want to print all the values stored in the nodes of the tree.
- In what order should we print them?

Traversing a Binary Tree

- Ways to print a 3 node tree:



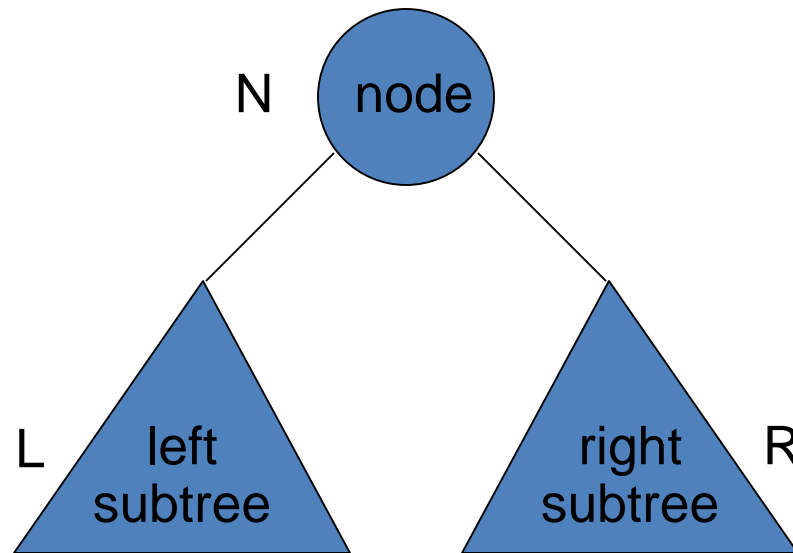
(4, 14, 15), (4,15,14)

(14,4,15), (14,15,4)

(15,4,14), (15,14,4)

Traversing a Binary Tree

- In case of the general binary tree:



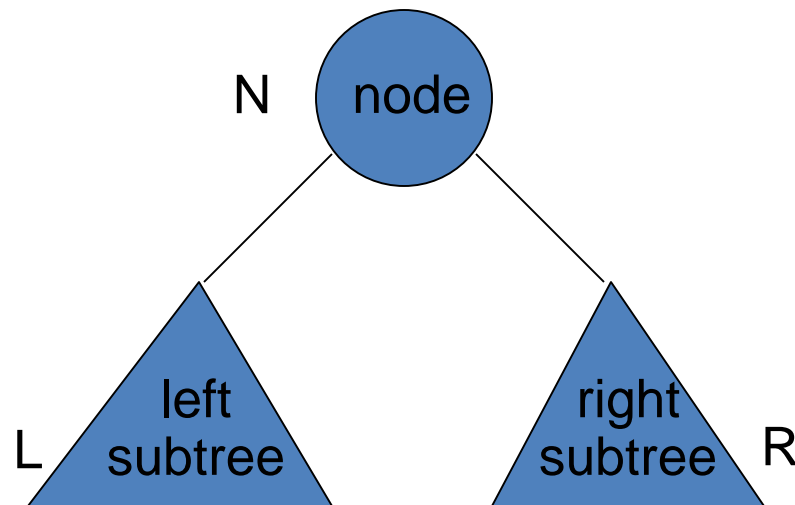
(L,N,R)

(L,R,N)

(N,L,R),

Traversing a Binary Tree

- Three common ways



Preorder: (N,L,R)

Inorder: (L,N,R)

Postorder: (L,R,N)



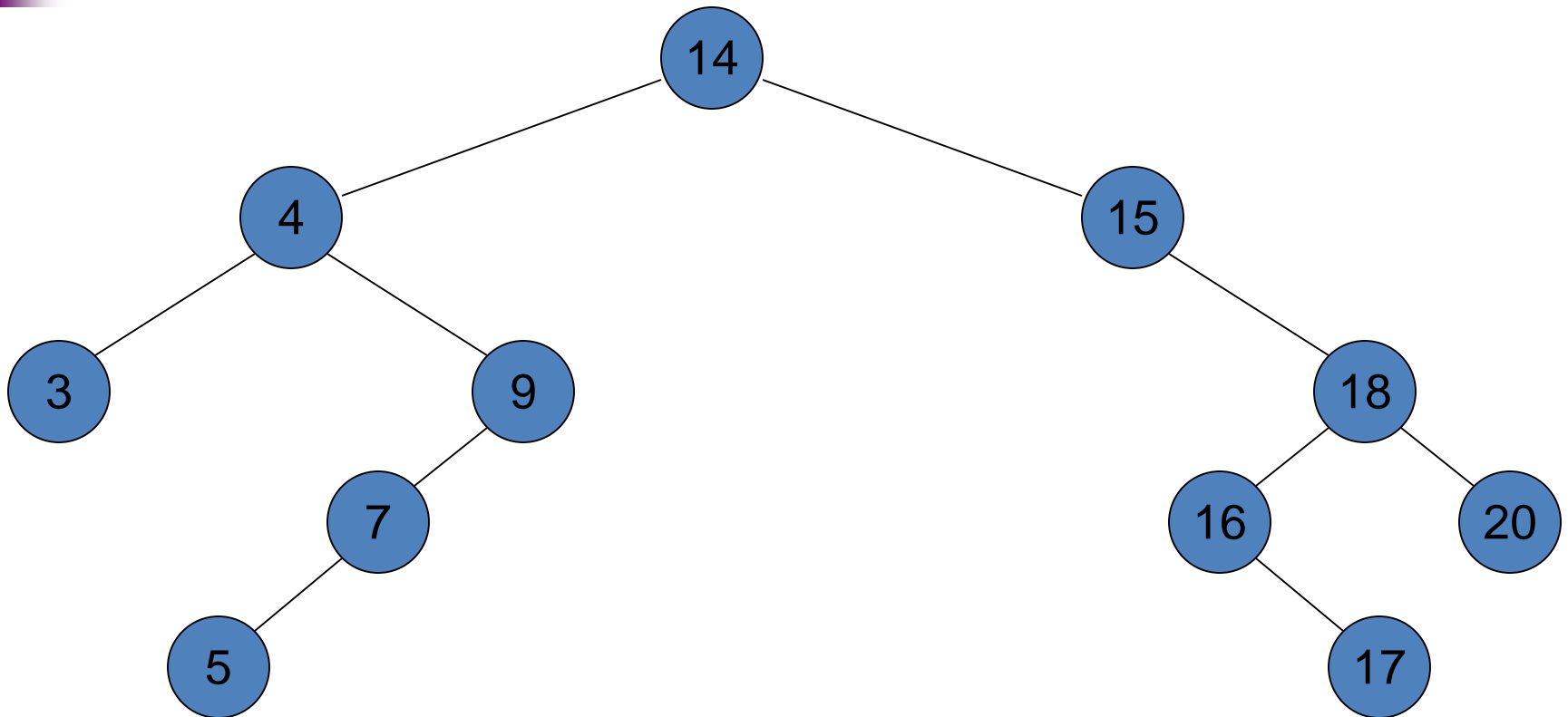
Traversing a Binary Tree

```
cout << "inorder: ";    preorder( root );  
cout << "inorder: ";    inorder( root );  
cout << "postorder: ";  postorder( root );
```

Traversing a Binary Tree

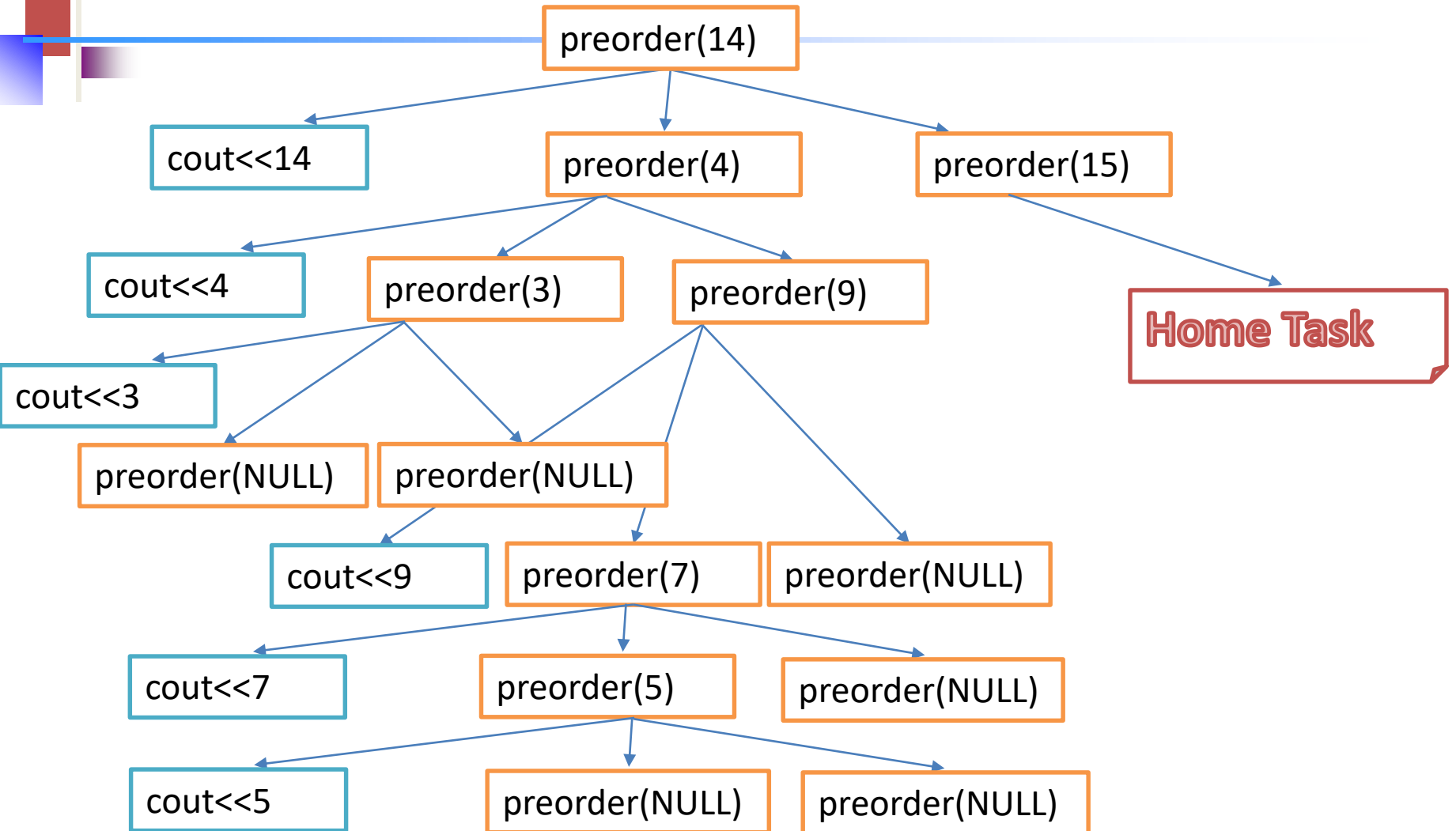
```
void preorder(TreeNode* treeNode)
{
    if( treeNode != NULL )
    {
        cout << (treeNode->getInfo()) << " ";
        preorder(treeNode->getLeft());
        preorder(treeNode->getRight());
    }
}
```


Traversing a Binary Tree



Preorder: 14 4 3 9 7 5 15 18 16 17 20

Preorder Traversal



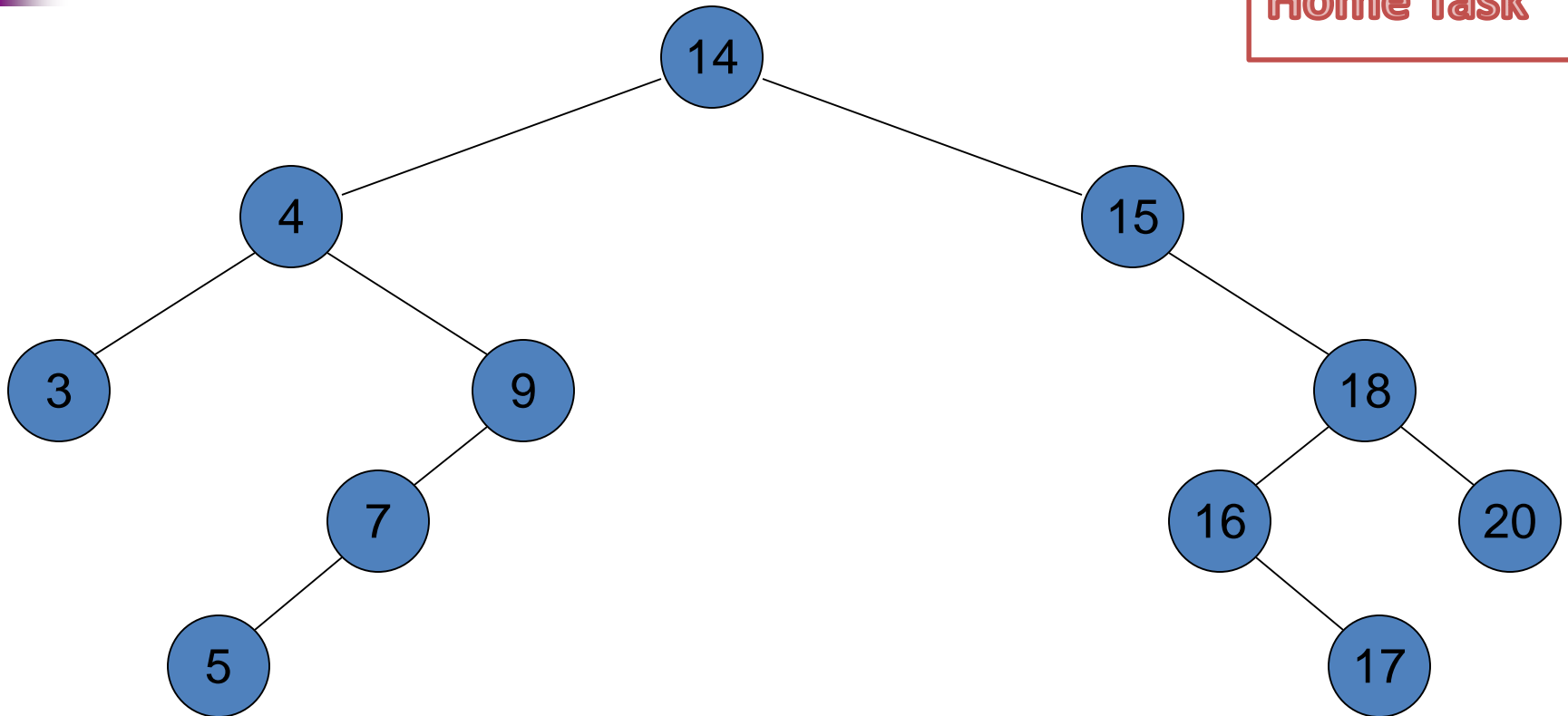
Preorder: 14 4 3 9 7 5 15 18 16 17 20

Traversing a Binary Tree

```
void postorder(TreeNode* treeNode)
{
    if( treeNode != NULL )
    {
        postorder(treeNode->getLeft());
        postorder(treeNode->getRight());
        cout << (treeNode->getInfo())<<" ";
    }
}
```

Traversing a Binary Tree

Home Task



Postorder:

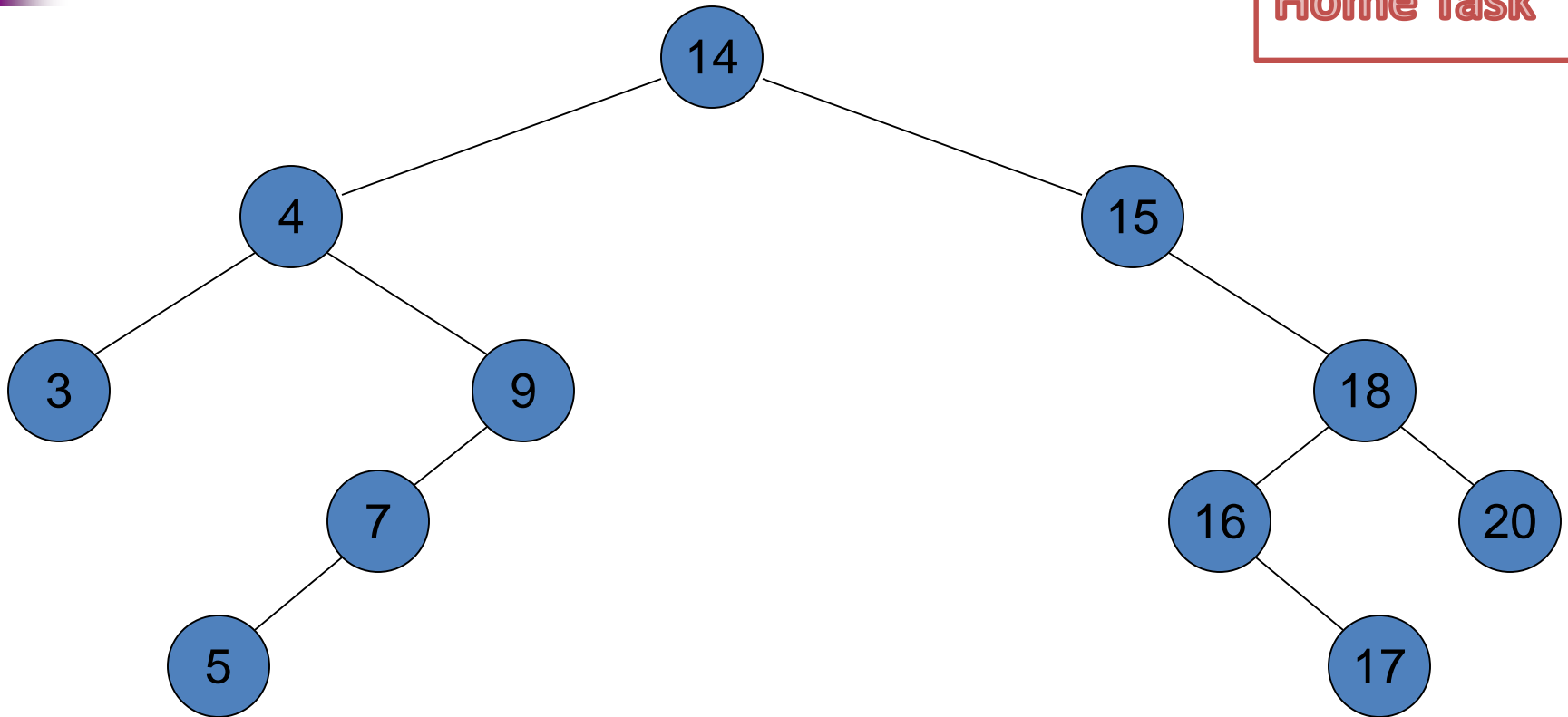


Traversing a Binary Tree

```
void inorder(TreeNode* treeNode)
{
    if( treeNode != NULL )
    {
        inorder(treeNode->getLeft());
        cout << (treeNode->getInfo())<<" ";
        inorder(treeNode->getRight());
    }
}
```

Traversing a Binary Tree

Home Task



Inorder: