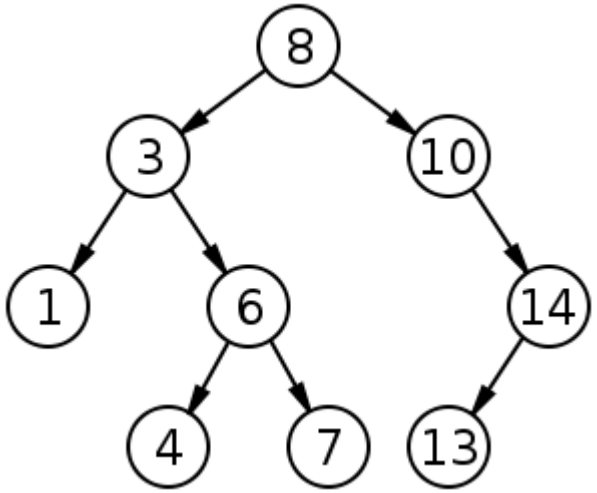
Data Structures & Algorithms

Tree

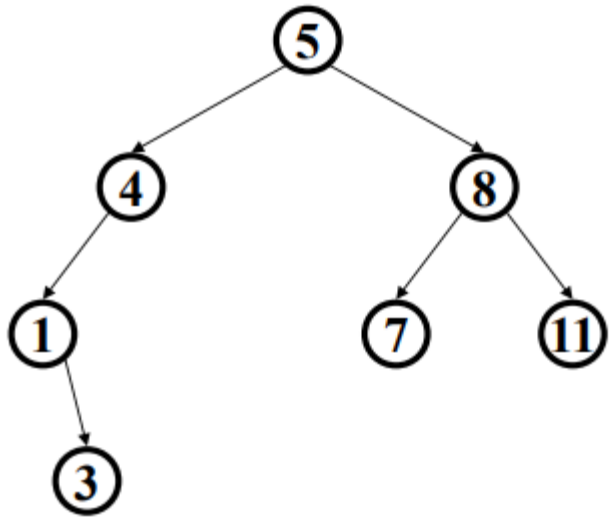
Binary Search Tree (BST)

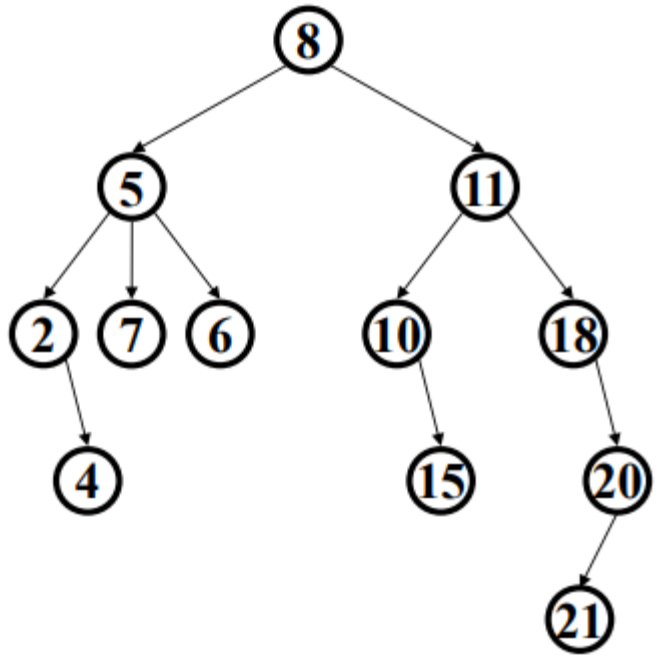
A binary search tree (BST) is a binary tree whose internal nodes store a key greater than all the keys in the node's left subtree and less than those in its right subtree.
In other words, the key in each node is greater than the key stored in the left sub-tree, and less than to any key stored in the right sub-tree.
We assume that keys are unique.



Binary Search Tree (BST)

Are these BSTs?





BST Operations

Traversal
Searching
Insertion
Deletion

Traversal Binary Search Tree (BST)

A traversal is an order for visiting all the nodes of a tree.

There are mainly three ways to traverse a BST:

- Inorder Traversal
- Preorder Traversal
- Postorder Traversal

Binary Search Tree Traversals

- Inorder Traversals
  - Visit nodes in the order
    - Left-Root-Right
- Preorder Traversal
  - Visit nodes in the order
    - Root-Left-Right
- Postorder Traversal
  - Visit nodes in
    - Left-Right-Root

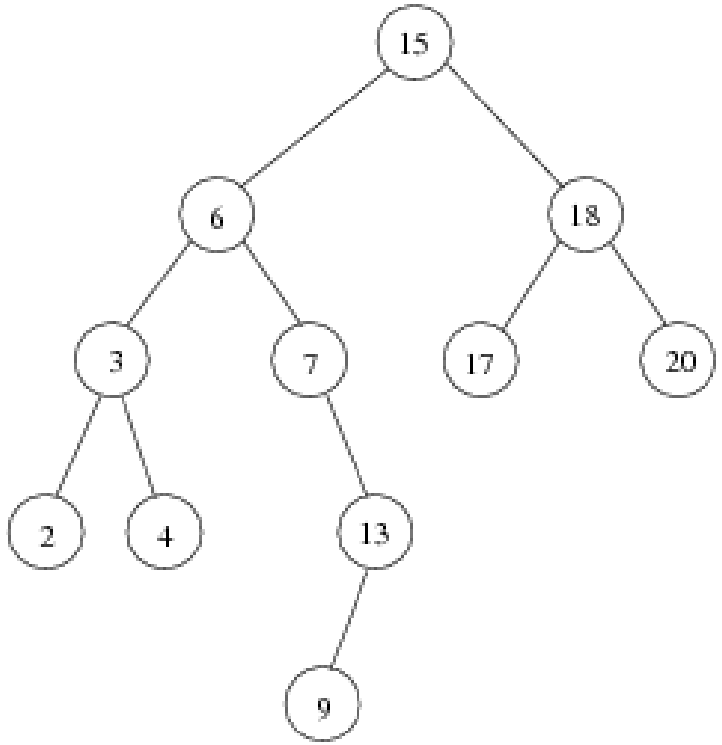
Inorder Traversal

Visit the nodes in the left subtree, then visit the root of the tree, then visit the nodes in the right subtree

```
Inorder(root)
{
   if (root != NULL)  {
     Inorder(root->left);
     print(root->data);
     Inorder(root->right);
  }
}
```

Inorder Traversal

Give an Inorder traversal of the following BST



Inorder: 2, 3, 4, 6, 7, 9, 13, 15, 17, 18, 20

Preorder Traversal

Visit the root of the tree first, then visit the nodes in the left subtree, then visit the nodes in the right subtree

```
Preorder(root)
{
   if (root != NULL)  {
     print(root->data);
     Preorder(root->left);
     Preorder(root->right);
  }
}
```

Postorder Traversal

Visit the nodes in the left subtree first, then visit the nodes in the right subtree, then visit the root of the tree

```
Postorder(root)
{
  if (root != NULL)  {
     Postorder(root->left);
     Postorder(root->right);
     print(root->data);
  }
}
```

Binary Search Tree (BST) Traversal

Preorder traversal of the given BST
Preorder: 15, 6, 3, 2, 4, 7, 13, 9, 18, 17, 20

Posteorder traversal of the given BST
Postorder: 2, 4, 3, 9, 13, 7, 6, 17, 20, 18, 15

Construction of BST from its Traversal

Given:

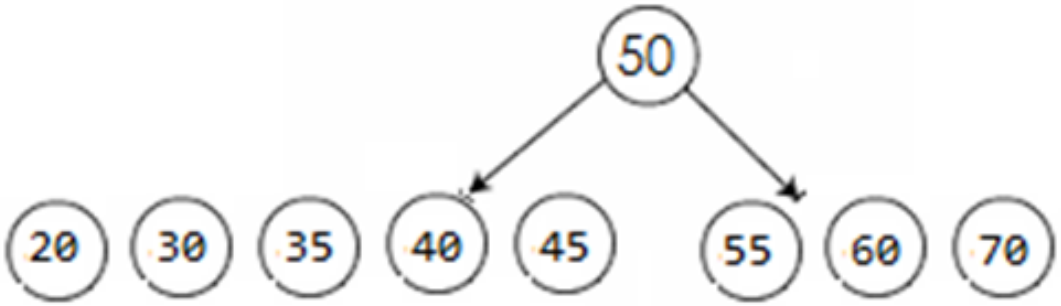
Post order : 20, 35, 30, 45, 40, 55, 70, 60, 50

In order : 20, 30, 35, 40, 45, 50, 55, 60, 70

Construct the BST and find its Pre order traversal sequence

```
inOrder     = 20, 30, 35, 40, 45, 50, 55, 60, 70

postOrder   = 20, 35, 30, 45, 40, 55, 70, 60, 50
```



Ans: Pre order is : 50, 40, 30, 20, 35, 45, 60, 55, 70

Construction of BST from its Traversal