Chapter 6: Check if a tree is BST or not

Section 6.1: Algorithm to check if a given binary tree is BST

A binary tree is BST if it satisfies any one of the following condition:

1. It is empty
2. It has no subtrees
3. For every node x in the tree all the keys (if any) in the left sub tree must be less than key(x) and all the keys (if any) in the right sub tree must be greater than key(x).

So a straightforward recursive algorithm would be:

```
is_BST(root):
  if root == NULL:
   return true

  // Check values in left subtree
  if root->left != NULL:
    max_key_in_left = find_max_key(root->left)
    if max_key_in_left > root->key:
        return false

  // Check values in right subtree
  if root->right != NULL:
    min_key_in_right = find_min_key(root->right)
    if min_key_in_right < root->key:
        return false

  return is_BST(root->left) && is_BST(root->right)
```

The above recursive algorithm is correct but inefficient, because it traverses each node mutiple times.

Another approach to minimize the multiple visits of each node is to remember the min and max possible values of the keys in the subtree we are visiting. Let the minimum possible value of any key be `K_MIN` and maximum value be `K_MAX`. When we start from the root of the tree, the range of values in the tree is `[K_MIN,K_MAX]`. Let the key of root node be `x`. Then the range of values in left subtree is `[K_MIN,x)` and the range of values in right subtree is `(x,K_MAX]`. We will use this idea to develop a more efficient algorithm.

```
is_BST(root, min, max):
    if root == NULL:
        return true

    // is the current node key out of range?
    if root->key < min || root->key > max:
        return false

    // check if left and right subtree is BST
    return is_BST(root->left,min,root->key-1) && is_BST(root->right,root->key+1,max)
```

It will be initially called as:

```
is_BST(my_tree_root,KEY_MIN,KEY_MAX)
```

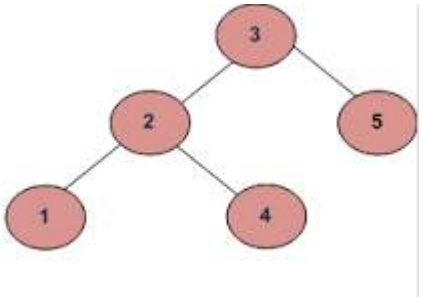
Another approach will be to do inorder traversal of the Binary tree. If the inorder traversal produces a sorted sequence of keys then the given tree is a BST. To check if the inorder sequence is sorted remember the value of

previously visited node and compare it against the current node.

Section 6.2: If a given input tree follows Binary search tree property or not

For example

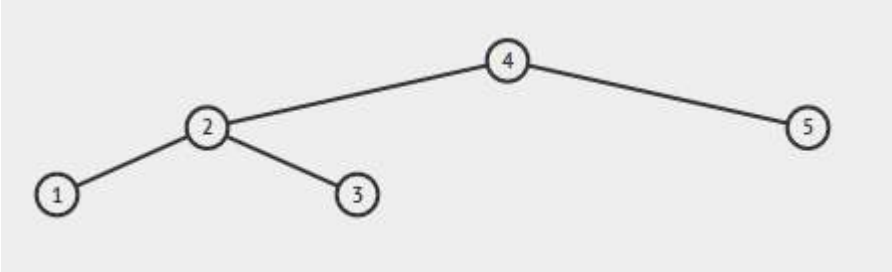
if the input is:



Output should be false:

As 4 in the left sub-tree is greater than the root value(3)

If the input is:



Output should be true