17. Binary search tree

A binary search tree is a node. It is a based binary-tree data structure which has the following properties

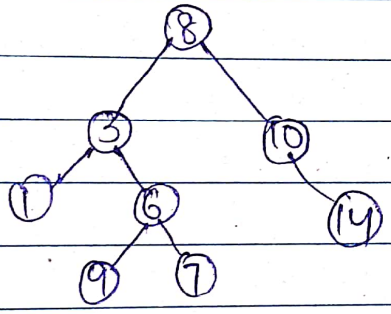
→ The left subtree of a node contains only nodes with keys lesser than the node's key

→ The right subtree of a node contains only nodes with keys greater than the node's key

→ The left and right subtree each must also be a binary search tree.

Example:-

Let the sample tree be



The search element is 4.

When the root node is not equal to the search number and number < root.data

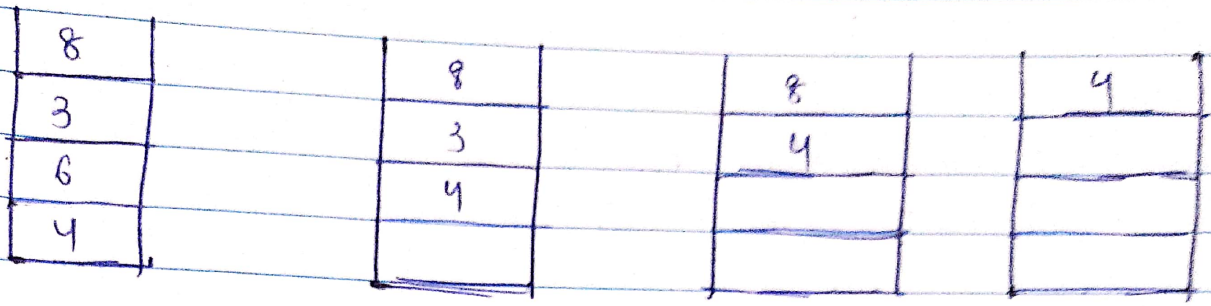
Traverse through the leftsubtree of the node

The next node-data is ③ as 4 > root.data(3) traverse through the right subtree

The next node data is 6. where 4 < 6. Hence traverse through the left subtree of 6

The leaf node data is same as the pivot number

Hence search is haulted and declared that the pivot element is found.



Code:

```
def search(root, Key):
    if root is None or root.val == Key
        return root
    if root.val < Key:
        return search(root.right, Key)
    return search(root.left, Key)
```