Binary Search Tree:

Insert a Node to Binary Search Tree:

To insert a node based on Binary tree Search we have to find a proper location that which we can place the node there:

So, Node class is:

```
class BinarySearchTree:
    def __init__(self,data):
        self.data = data
        self.rightChild = None
        self.leftChild = None
```

at first, we have to check if the root Node data that we gave insert function is Note None:

```
def insert(rootNode,Node):
    if rootNode.data == None:
        rootNode.data = Node
```

if it is we have to put the given value as the data of the root Node – so this is the first case scenario of inserting a node into a Binary search tree

then the second case is that we have a root node which it has at least one child:

now we go on and compare the given value to the Binary search's tree data and if the value was bigger than the Node's data we go and seek through the right-sub-Tree else we repeat the algorithm for the left-sub-Tree and we do this process by calling our function recursively:

```
def insert(rootNode,Node): #----> O(n) time complexity and O(log n) space complexity
    if rootNode.data == None:
        rootNode.data = Node
    elif Node <= rootNode.data:
        if rootNode.leftChild is None:
            rootNode.leftChild = BinarySearchTree(Node) #---O(n/2) time complexity
        else:
            insert(rootNode.leftChild,Node)
    else:
        if rootNode.rightChild is None:
            rootNode.rightChild = BinarySearchTree(Node) #----O(n/2) time complexity
        else:
            insert(rootNode.rightChild,Node)
```