```
class Node:
    def __init__(self, data):
        self.data = data
        self.left = None
        self.right = None
        self.count = 1

class BinarySearchTree:
    def __init__(self):
        self.root = None
        self.rank = 0

    def insert(self, data):
        if self.root is None:
            self.root = Node(data)
        else:
            self.insertNode(data, self.root)

    def insertNode(self, data, node):
        if data < node.data:
            if node.left:
                self.insertNode(data, node.left)
            else:
                node.left = Node(data)
        elif data > node.data:
            if node.right:
                self.insertNode(data, node.right)
            else:
                node.right = Node(data)
        else:
            node.count += 1

    def inorder_rank(self, number):
        self.rank = -1
        self.counter = 0
        self.inorder_rank_helper(self.root, number)
        return self.rank

    # recursively visit the tree.
    def inorder_rank_helper(self, node, number):
        if node is not None:
            self.inorder_rank_helper(node.left, number)
            if node.data == number:
                self.rank = self.counter
            self.counter += node.count
            self.inorder_rank_helper(node.right, number)


if __name__ == '__main__':
    bst = BinarySearchTree()
    bst.insert(16)
    bst.insert(12)
    bst.insert(15)
    bst.insert(80)
```

```
bst.insert(32)
bst.insert(90)
print(bst.inorder_rank(16))
```