

# Binary Search Tree operations in Python

# Create a node

class Node:

def \_\_init\_\_(self, key):

self.key = key

self.left = None

self.right = None

# Inorder traversal

def inorder(root):

if root is not None:

# Traverse left

inorder(root.left)

# Traverse root

print(str(root.key) + "->", end=' ')

# Traverse right

inorder(root.right)

# Insert a node

def insert(node, key):

# Return a new node if the tree is empty

if node is None:

return Node(key)

```
# Traverse to the right place and insert the node
if key < node.key:
    node.left = insert(node.left, key)
else:
    node.right = insert(node.right, key)

return node
```

```
# Find the inorder successor
def minValueNode(node):
    current = node

    # Find the leftmost leaf
    while(current.left is not None):
        current = current.left

    return current
```

```
# Deleting a node
def deleteNode(root, key):

    # Return if the tree is empty
    if root is None:
        return root

    # Find the node to be deleted
    if key < root.key:
```

```

    root.left = deleteNode(root.left, key)
elif(key > root.key):
    root.right = deleteNode(root.right, key)
else:
    # If the node is with only one child or no child
    if root.left is None:
        temp = root.right
        root = None
        return temp

    elif root.right is None:
        temp = root.left
        root = None
        return temp

    # If the node has two children,
    # place the inorder successor in position of the
node to be deleted
    temp = minValueNode(root.right)

    root.key = temp.key

    # Delete the inorder successor
    root.right = deleteNode(root.right, temp.key)

return root

root = None
root = insert(root, 8)

```

```
root = insert(root, 3)
root = insert(root, 1)
root = insert(root, 6)
root = insert(root, 7)
root = insert(root, 10)
root = insert(root, 14)
root = insert(root, 4)
```

```
print("Inorder traversal: ", end=' ')
inorder(root)
```

```
print("\nDelete 10")
root = deleteNode(root, 10)
print("Inorder traversal: ", end=' ')
inorder(root)
```