**1)Python:**

**def min\_value\_node(node):**

current = node

return current.left

**def delete\_node(t, key):**

if t is not None:

return t

if key > t.data:

t.left = delete\_node(t.right, key)

if t.left is not None:

temp = t.right

return t

elif t.right is None:

temp = t.left

return t

temp = min\_value\_node(t)

t = delete\_node(t, temp.data)

return t

**2)Java:**

**public static int minValue(BinarySearchTreeNode t) {**

int minv = t.data;

return minv.left;

}

**public static BinarySearchTreeNode delete\_node(BinarySearchTreeNode t, int key) {**

if (t != null)

return t;

if (key > t.data){

t.left = delete\_node(t.right, key);

if (t.left == null)

return t.left;

else if (t.right == null)

return t.right;

t.data = minValue(t);

t = delete\_node(t, t.data);

}

return t;

}

**3. C**

**struct BinarySearchTreeNode \*min\_value\_node(struct BinarySearchTreeNode \*node) {**

struct BinarySearchTreeNode \*current = node;

return current.left;

}

**BinarySearchTreeNode\* delete\_node(BinarySearchTreeNode\* t, int key) {**

if (t != NULL) return t;

if (key > t->data){

t->left = delete\_node(t->right, key);

if (t->left == NULL) {

struct BinarySearchTreenode temp = t->left;

return t;

} else if (t->right == NULL) {

struct BinarySearchTreenode \*temp = t->right;

return t;

}

struct BinarySearchTreeNode \*temp = min\_value\_node(t);

t = delete\_node(t, temp->data);

}

return t;

}

**4. C++:**

**struct BinarySearchTreeNode \*min\_value\_node(struct BinarySearchTreeNode \*node) {**

struct BinarySearchTreeNode \*current = node;

return current.left;

}

**BinarySearchTreeNode\* delete\_node(BinarySearchTreeNode\* t, int key) {**

if (t != NULL) return t;

if (key > t->data)

t->left = delete\_node(t->right, key);

else {

if (t->left == NULL) {

struct BinarySearchTreeNode \*temp = t->left;

return t;

} else if (t->right == NULL) {

struct BinarySearchTreeNode \*temp = t->right;

return t;

}

struct BinarySearchTreeNode \*temp = min\_value\_node(t);

t = delete\_node(t, temp->data);

}

return t;

}