

Data Structures

Tree – Binary Search Tree

Team Emertxe



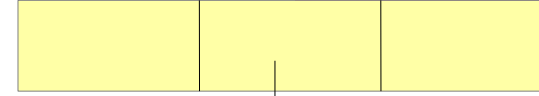
Binary Search Tree - Insert



Insert

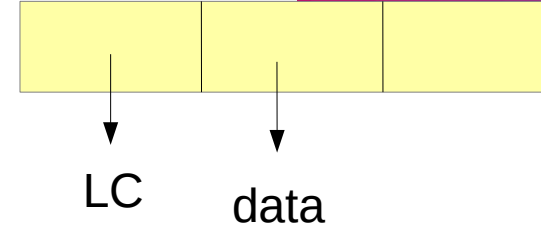


Insert

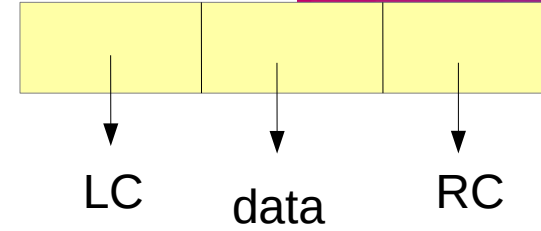


data

Insert

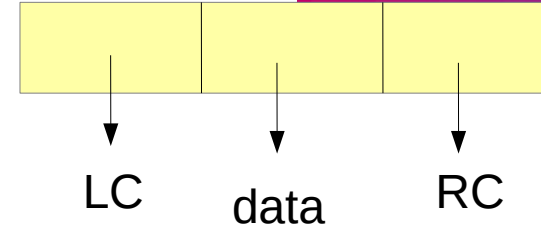


Insert



Insert

```
typedef struct node
{
    struct node * LC;
    int data;
    struct node * RC;
} tree_t;
```



Address of Left child

Item

Address of right child

Algorithm



bst_insert(root,data):

Input Specification:

root : Pointer that contains address of structure pointer (tree_t)
data : Item to be added

Algorithm



bst_insert(root,data):

Input Specification:

root : Pointer that contains address of structure pointer (tree_t)

data : Item to be added

Output Specification:

Status : e_true / e_false

Algorithm

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

If (flag = 1)
Parent->LC = new
Else
Parent->RC = new
return e_true

bst_insert(root,data)



30 , 20 , 50 , 10 , 55, 25, 52

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
    If (flag = 1)
        Parent->LC = new
    Else
        Parent->RC = new
    return e_true
```

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

```
        Parent->LC = new
```

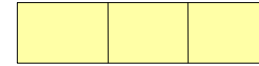
```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

```
        Parent->LC = new
```

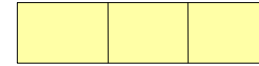
```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new ->LC =NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp -> data )
```

```
        temp = temp -> LC
```

```
        flag = 1
```

```
    Else if (data > temp -> data)
```

```
        temp = temp ->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

```
        Parent ->LC = new
```

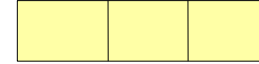
```
    Else
```

```
        Parent -> RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new

	30	
--	----	--

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag == 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new

NULL	30	
------	----	--

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new

NULL	30	
------	----	--

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
    If (flag == 1)
        Parent->LC = new
    Else
        Parent->RC = new
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new

NULL	30	NULL
------	----	------

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

new

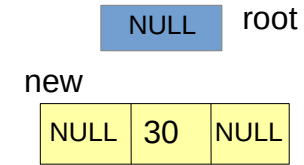
NULL	30	NULL
------	----	------

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

If (flag = 1)
Parent ->LC = new
Else
Parent -> RC = new
return e_true

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

If (flag = 1)
Parent ->LC = new
Else
Parent -> RC = new
return e_true

30 , 20 , 50 , 10 , 55, 25, 52

NULL root

new

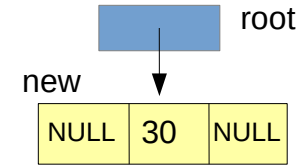
NULL	30	NULL
------	----	------

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

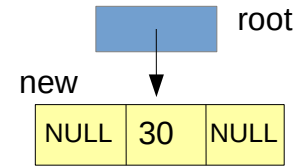
30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

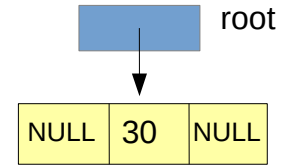
```
        Parent->LC = new
```

```
    Else
```

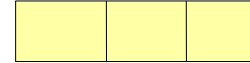
```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

```
        Parent->LC = new
```

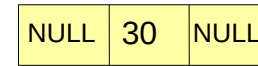
```
    Else
```

```
        Parent->RC = new
```

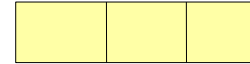
```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

root



new

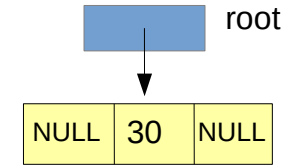


bst_insert(root,data)

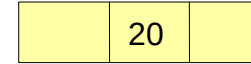
```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new

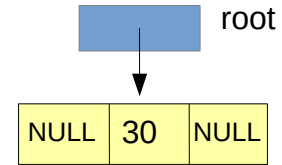


bst_insert(root,data)

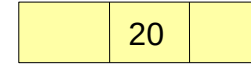
```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new

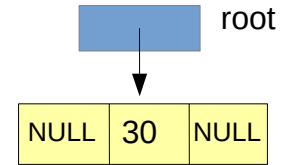


bst_insert(root,data)

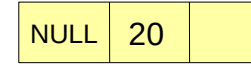
```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new

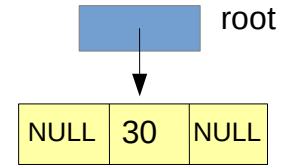


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

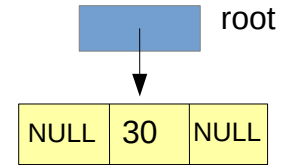


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new

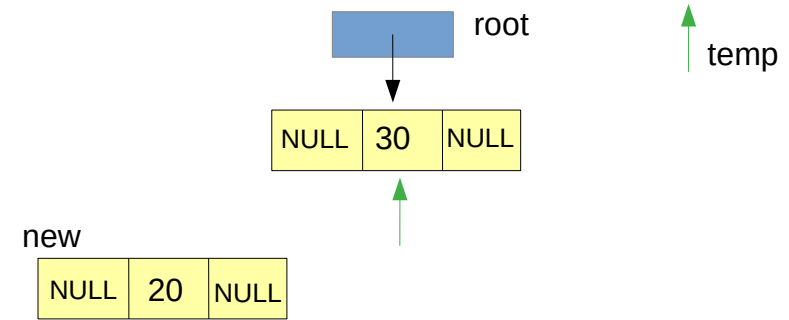


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

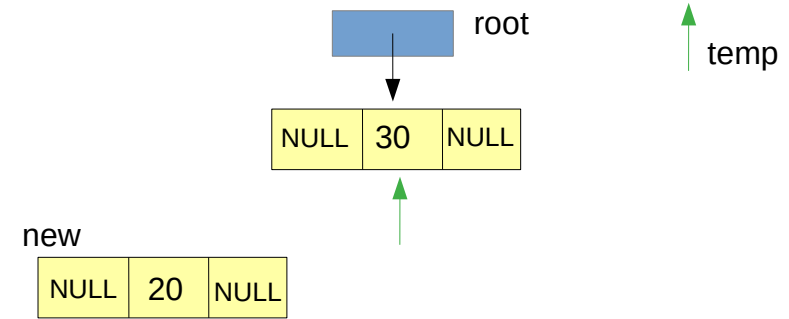


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

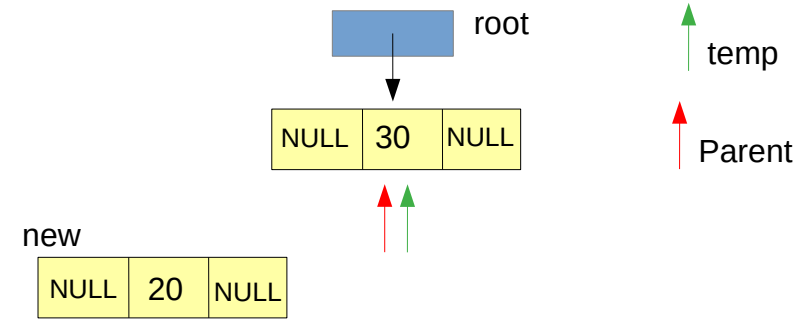


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
    flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52

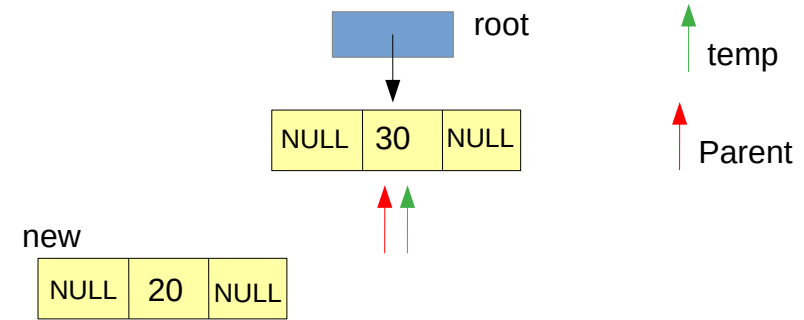


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
    flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

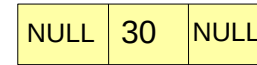
```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52

temp

NULL

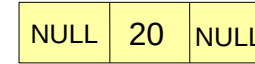
root



temp

Parent

new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

```
        Parent->LC = new
```

```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

temp

NULL

flag = 1

new

NULL	20	NULL
------	----	------

--

 root

NULL	30	NULL
------	----	------

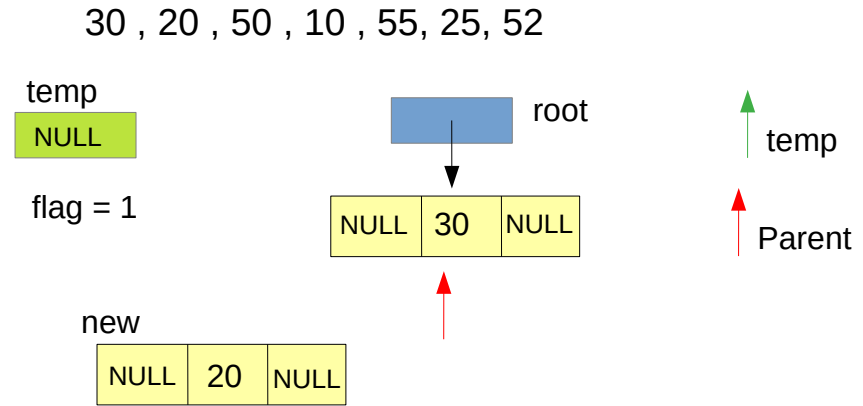
↑ temp

↑ Parent

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND

If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

If (flag = 1)

Parent ->LC = new

Else

Parent -> RC = new

return e_true

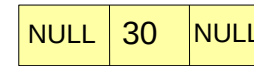
30 , 20 , 50 , 10 , 55 , 25 , 52

temp

NULL

flag = 1

new



temp

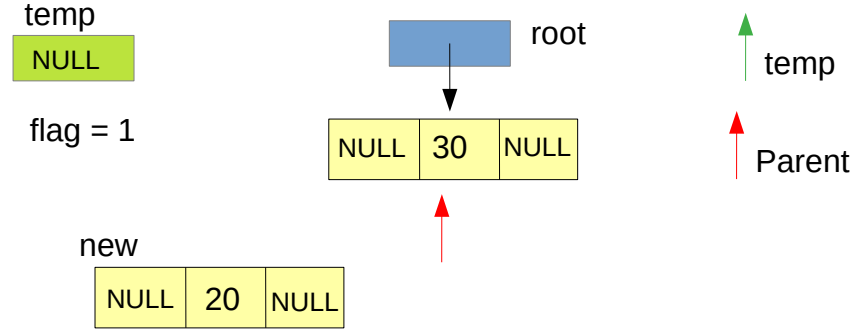
Parent

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52

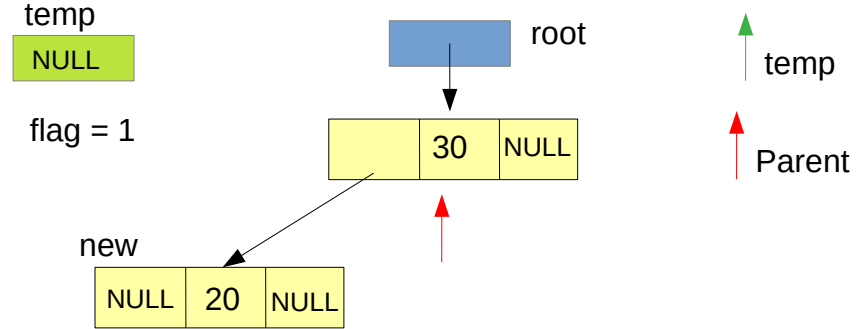


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52

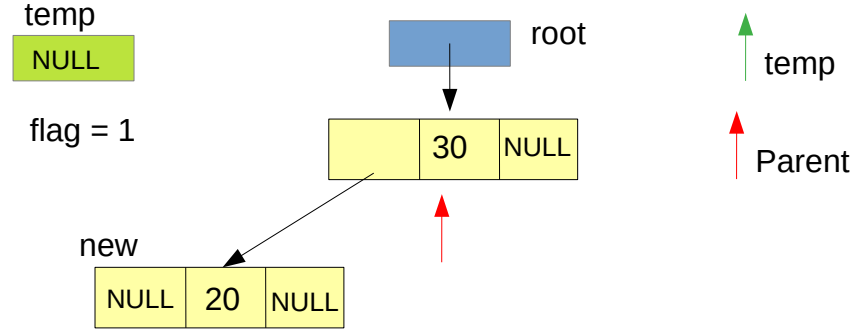


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
        Parent->LC = new
    Else
        Parent->RC = new
    return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52



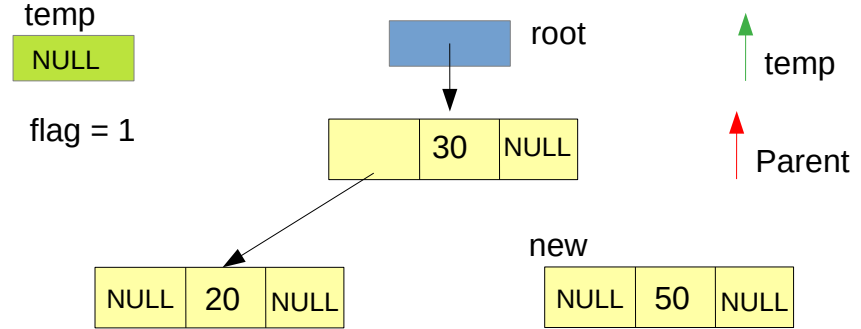
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

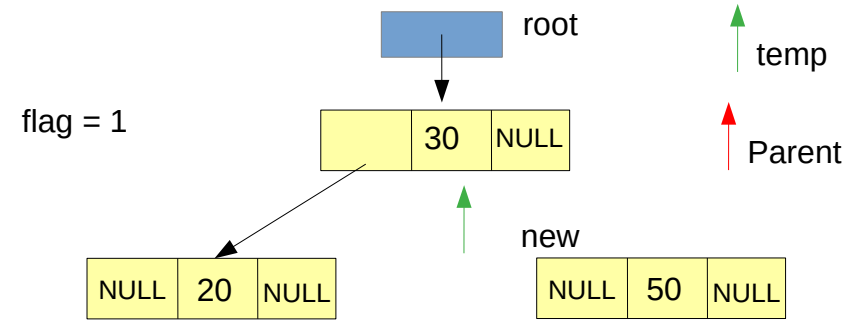


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



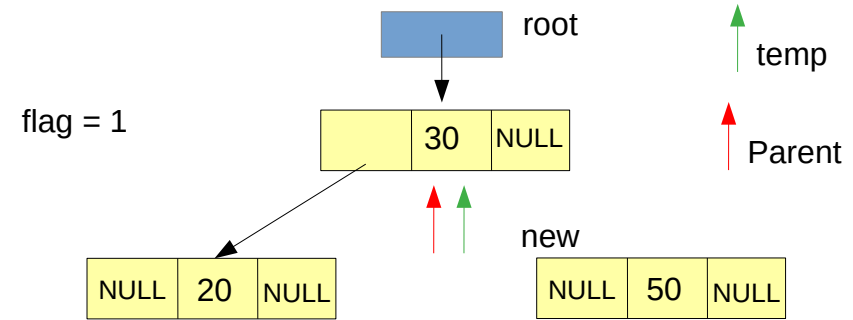
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



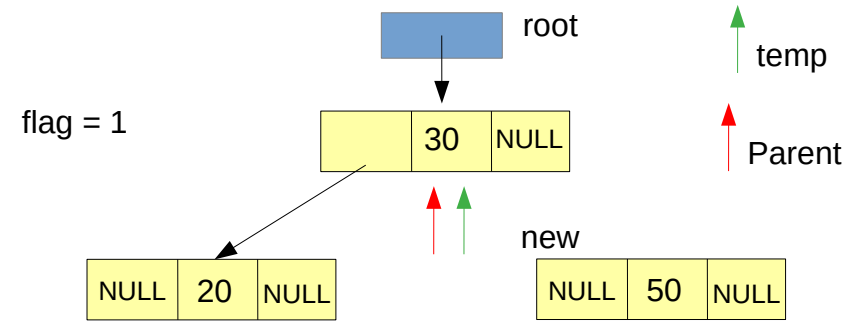
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



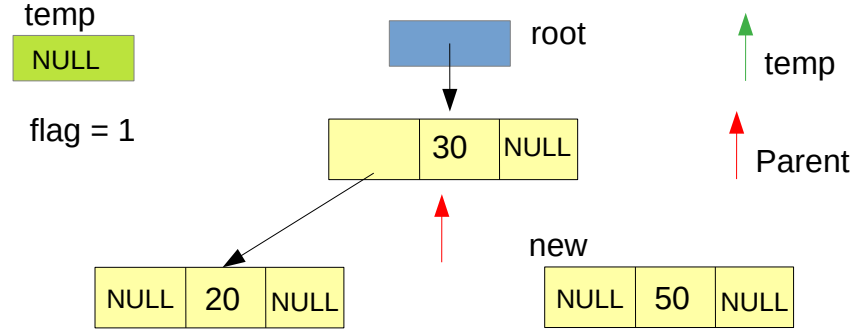
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52



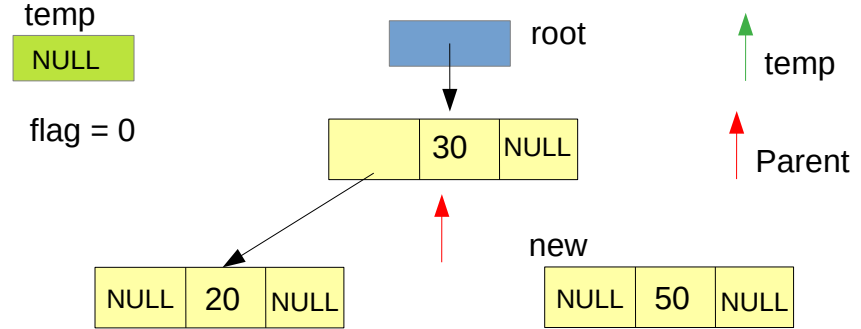
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

If (flag = 1)

Parent ->LC = new

Else

Parent -> RC = new

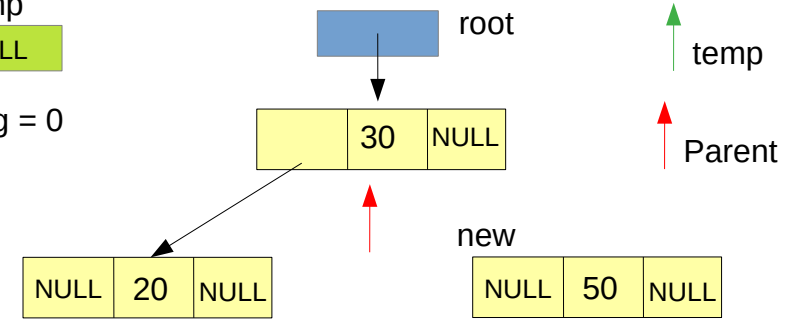
return e_true

30 , 20 , 50 , 10 , 55, 25, 52

temp

NULL

flag = 0



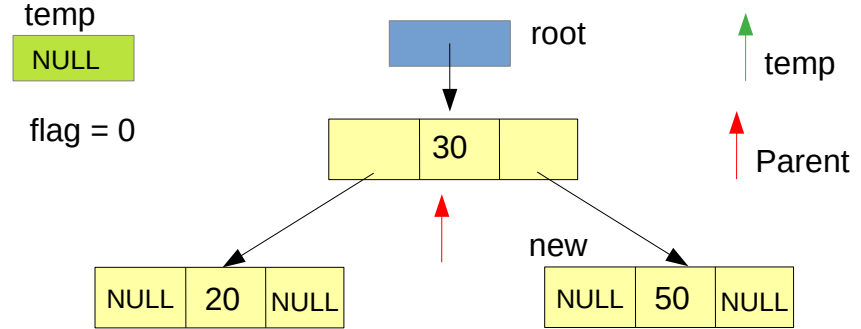
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



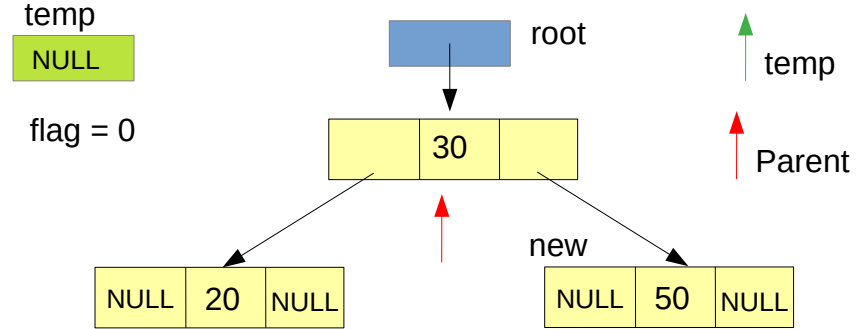
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new == NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root == NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

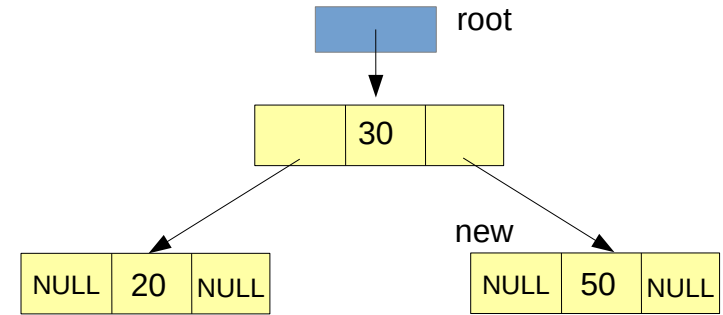
```
        Parent->LC = new
```

```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

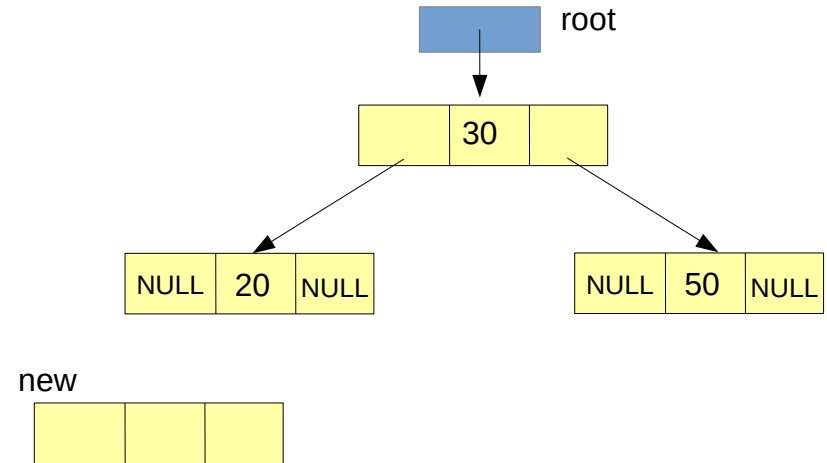
```
        Parent->LC = new
```

```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

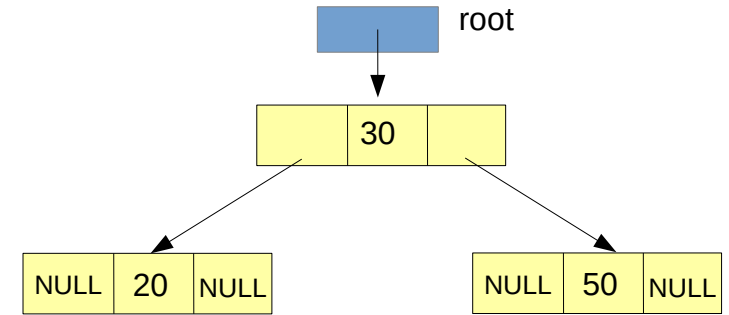
```
        Parent->LC = new
```

```
    Else
```

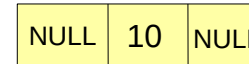
```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new

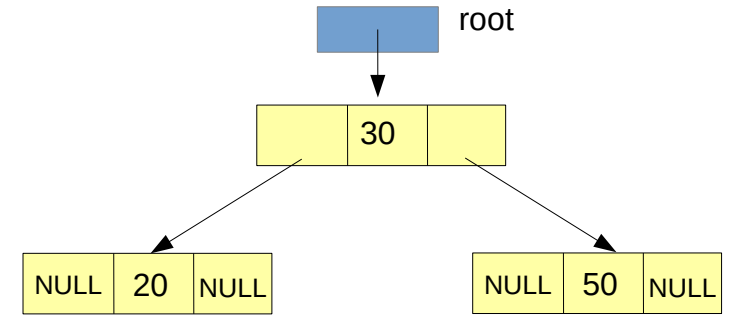


bst_insert(root,data)

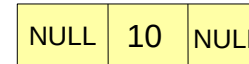
```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

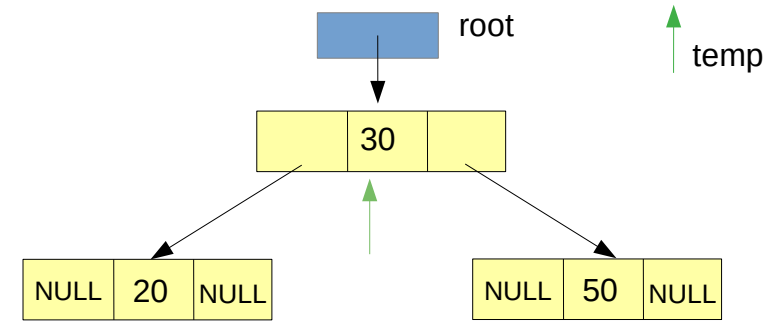
```
        Parent->LC = new
```

```
    Else
```

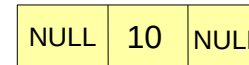
```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new



bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

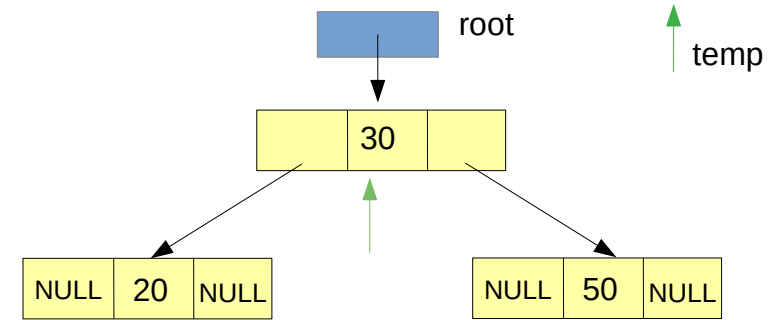
```
        Parent->LC = new
```

```
    Else
```

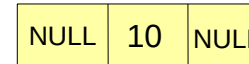
```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



new



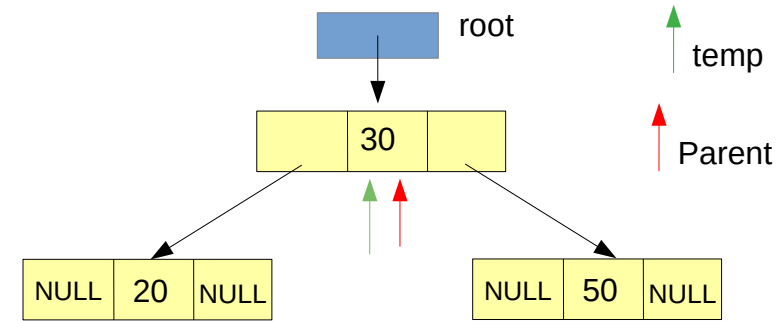
Data Structure – Binary search Tree

bst_insert(root,data)

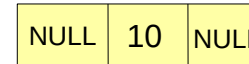
```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52



new



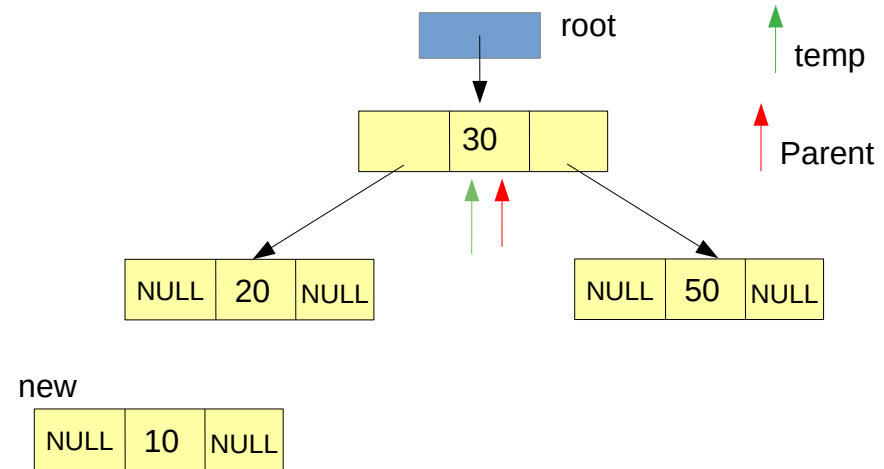
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55 , 25 , 52

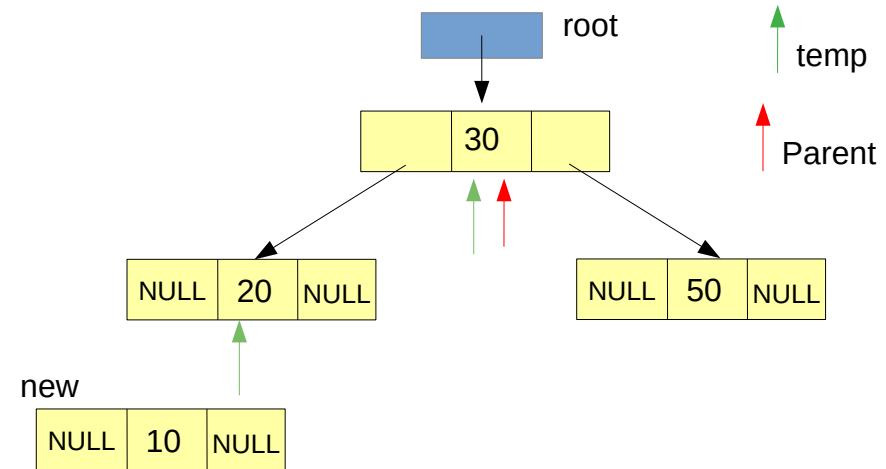


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



bst_insert(root,data)

```
return DUPLICATE_FOUND
```

```
return e_true
```



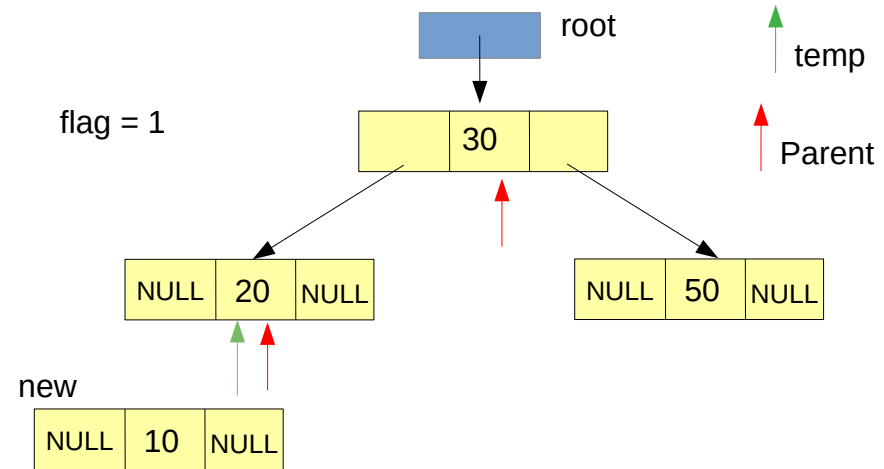
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    if( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



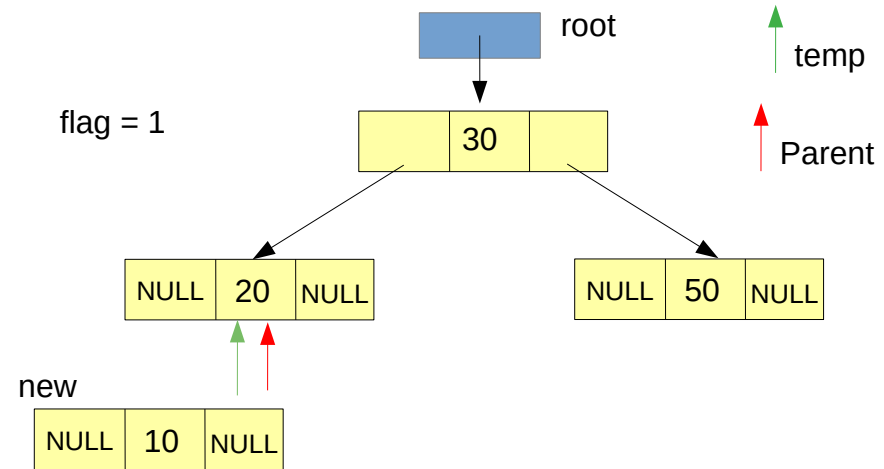
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

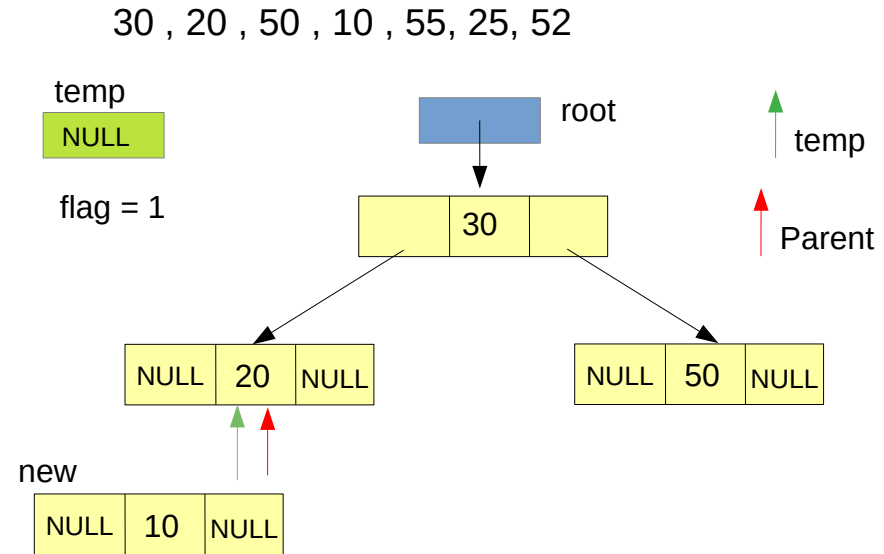


Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```



Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
```

```
if(new = NULL)
```

```
    return e_false
```

```
new->data = data
```

```
new->LC = NULL
```

```
new->RC = NULL
```

```
if(root = NULL)
```

```
    root = new
```

```
    return e_true
```

```
temp = root
```

```
while(temp != NULL)
```

```
    Parent = temp
```

```
    If( data < temp->data )
```

```
        temp = temp->LC
```

```
        flag = 1
```

```
    Else if (data > temp->data)
```

```
        temp = temp->RC
```

```
        flag = 0
```

```
    Else
```

```
        return DUPLICATE_FOUND
```

```
    If (flag = 1)
```

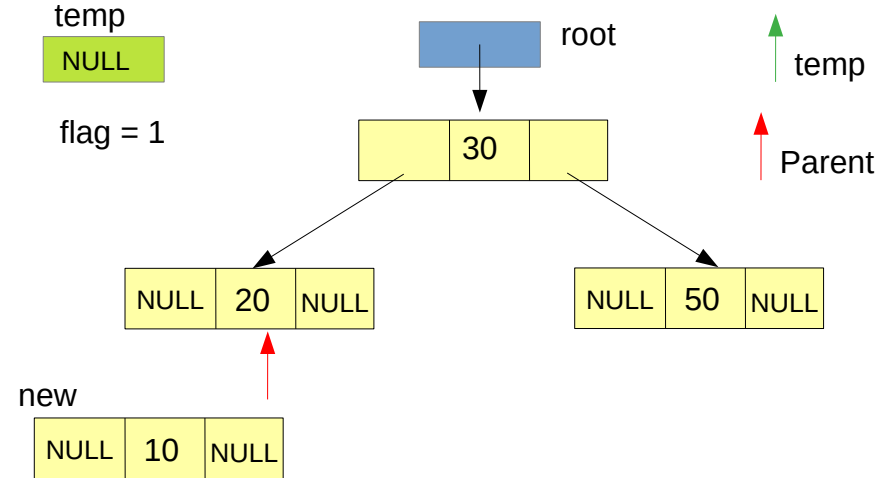
```
        Parent->LC = new
```

```
    Else
```

```
        Parent->RC = new
```

```
    return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



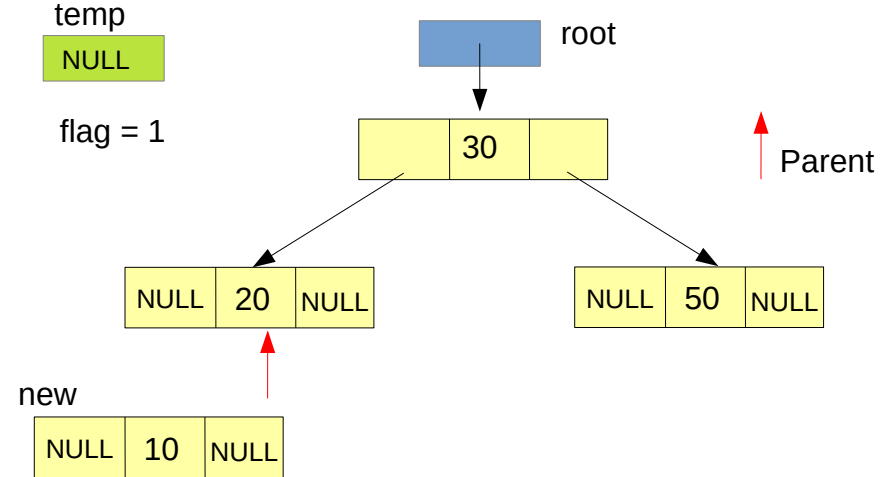
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

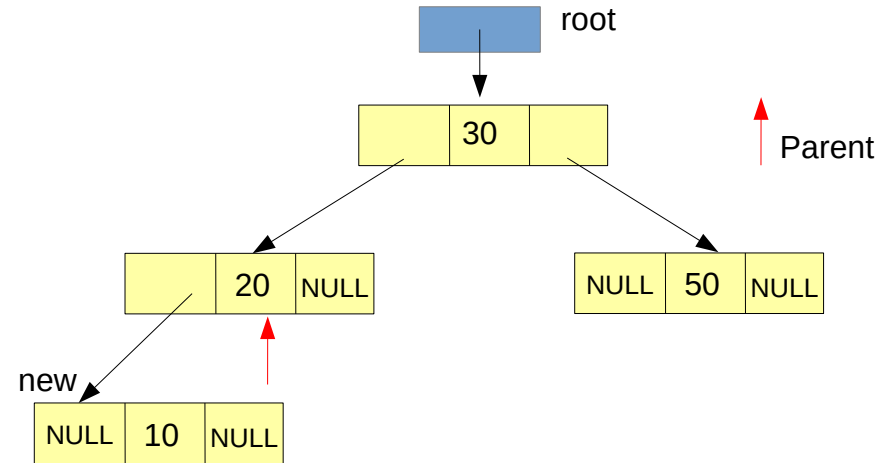


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



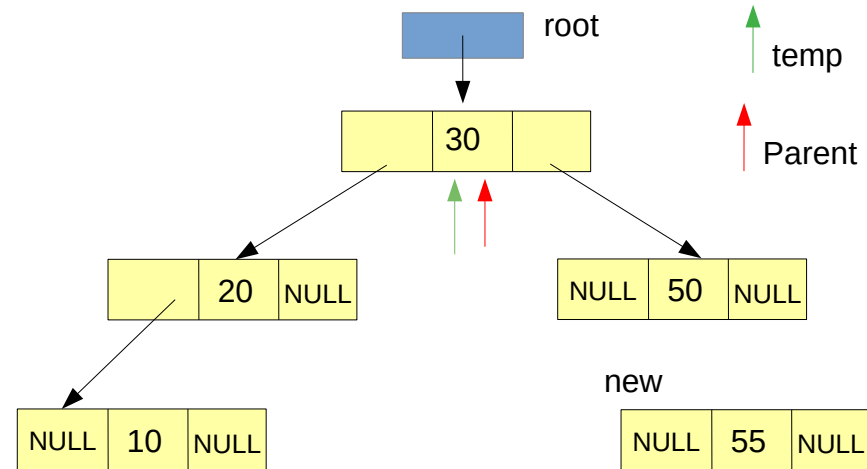
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52

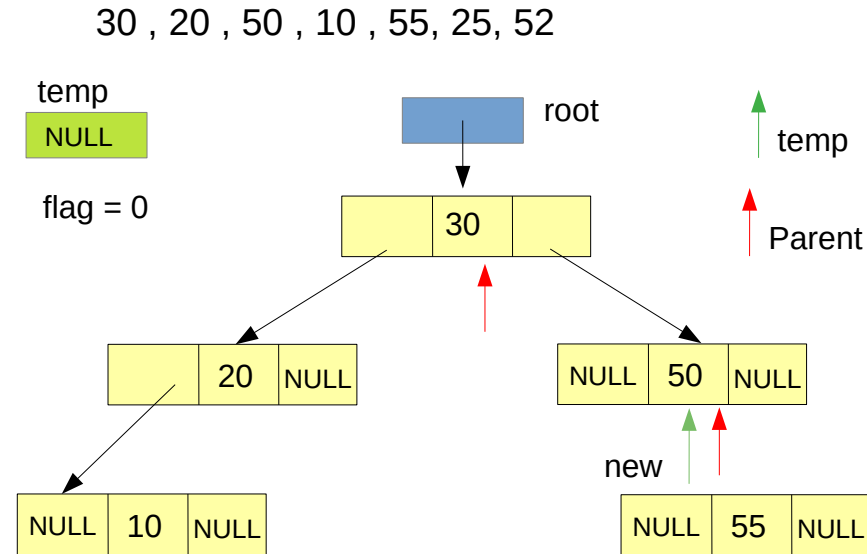


Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

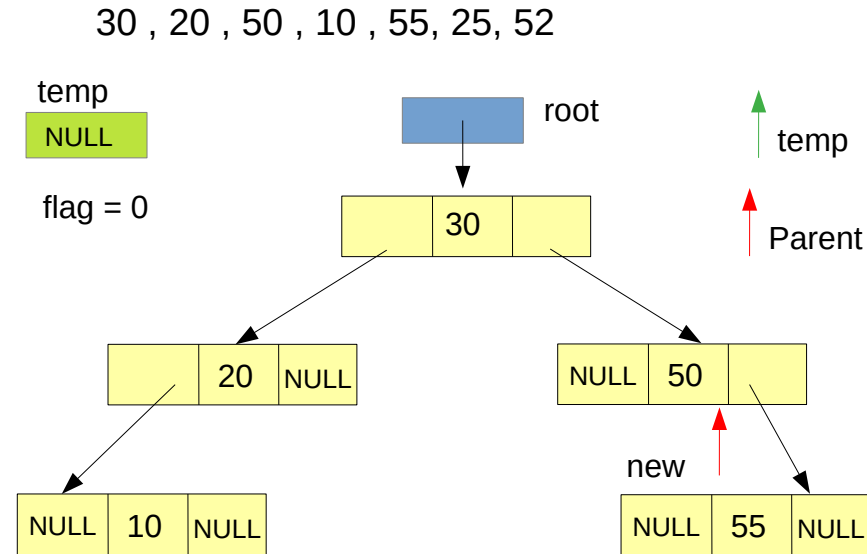


Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

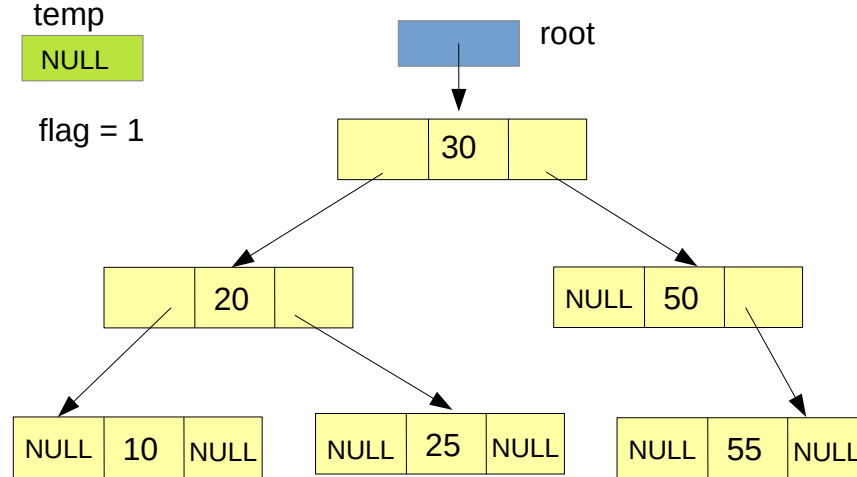


bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



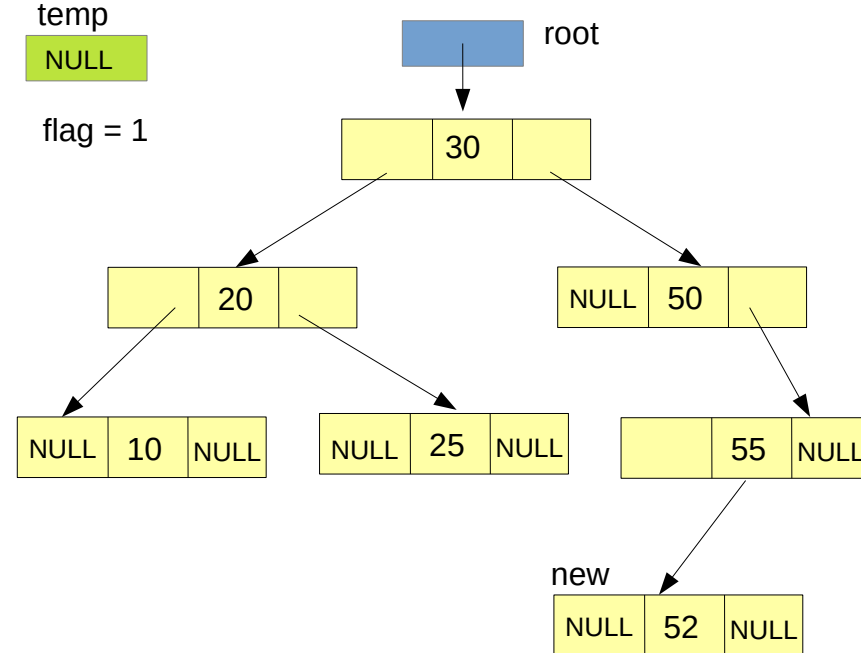
Data Structure – Binary search Tree

bst_insert(root,data)

```
new = Memalloc(sizeof(tree_t))
if(new = NULL)
    return e_false
new->data = data
new->LC = NULL
new->RC = NULL
if(root = NULL)
    root = new
    return e_true
temp = root
while(temp != NULL)
    Parent = temp
    If( data < temp->data )
        temp = temp->LC
        flag = 1
    Else if (data > temp->data)
        temp = temp->RC
        flag = 0
    Else
        return DUPLICATE_FOUND
```

```
If (flag = 1)
    Parent->LC = new
Else
    Parent->RC = new
return e_true
```

30 , 20 , 50 , 10 , 55, 25, 52



Code - `bst_insert(root,data)`

