Data Structures

# Tree – Binary Search Tree

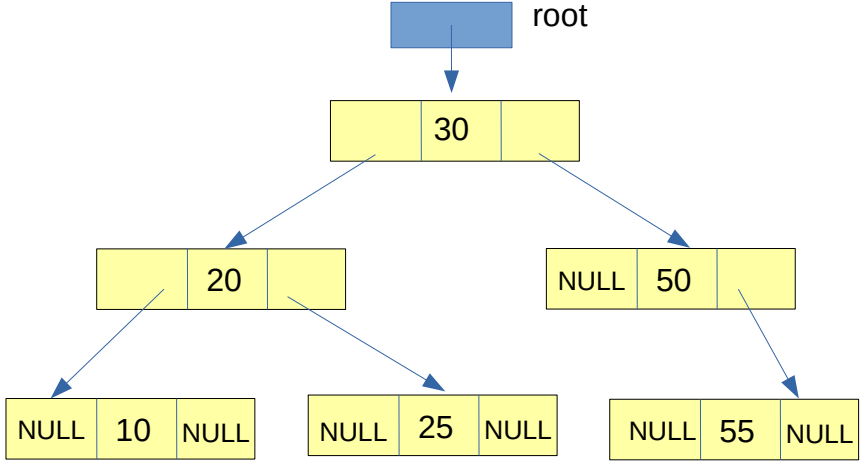
Team Emertxe

# Binary Search Tree -Inorder Traversal

# Inorder

Data Structure – Binary Search Tree -Traversal

# Inorder




Visit Left child

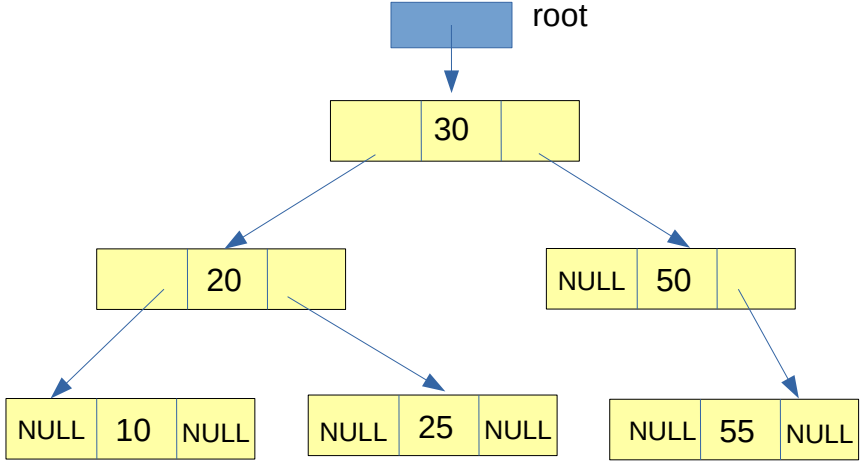
Visit Parent

Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder



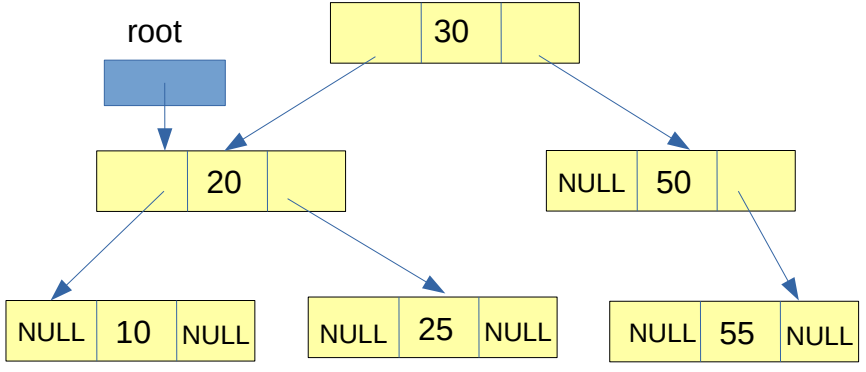

Visit Left child

Visit Parent

Visit Right child

EMERTXE

# Inorder



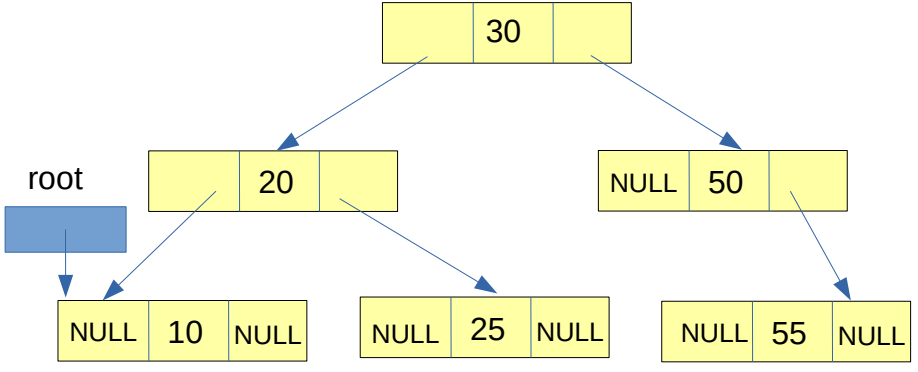

Visit Left child

Visit Parent

Visit Right child

EMERTXE

# Inorder




Visit Left child

Visit Parent

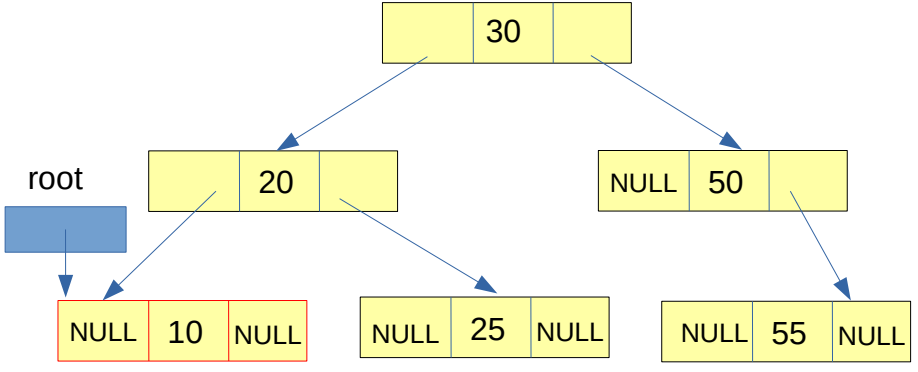
Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10




Visit Left child

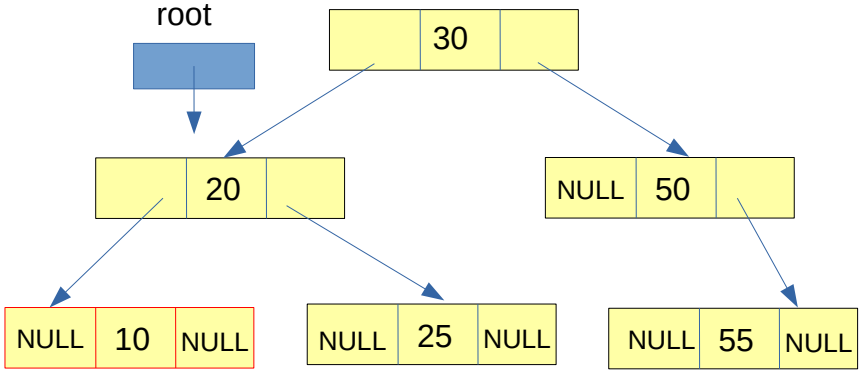
Visit Parent

Visit Right child

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20




Visit Left child

Visit Parent

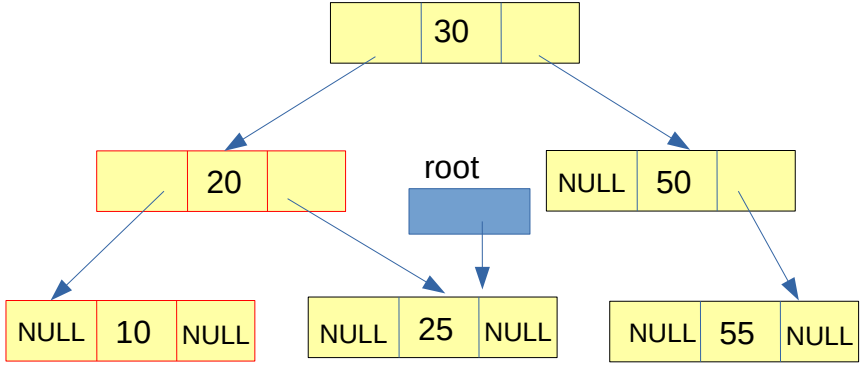
Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20 25




Visit Left child

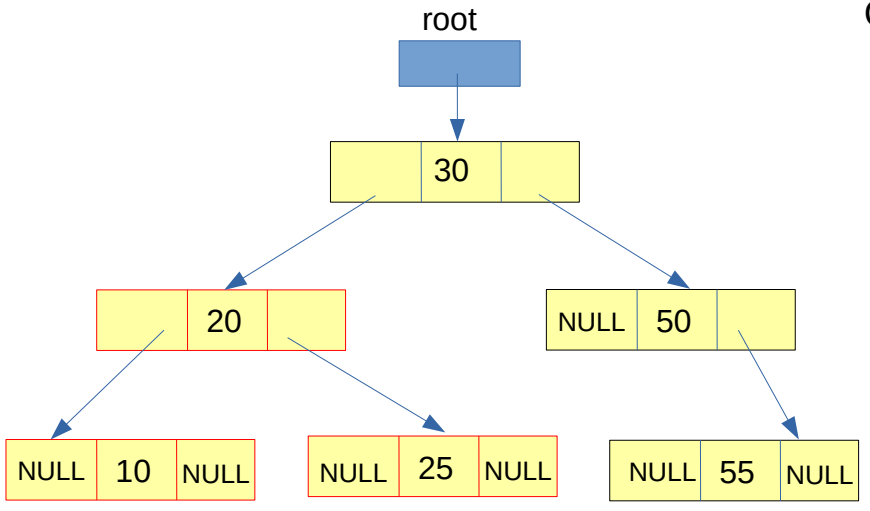
Visit Parent

Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder




OUTPUT = 10 20 25 30

Visit Left child

Visit Parent

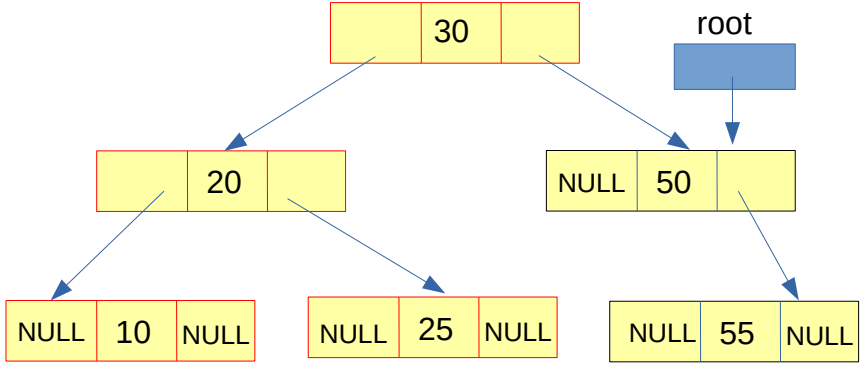
Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20 25 30




Visit Left child

Visit Parent

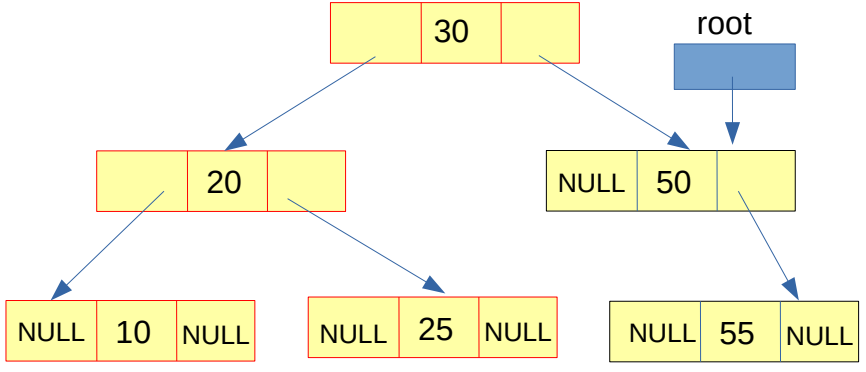
Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20 25 30 50




Visit Left child

Visit Parent

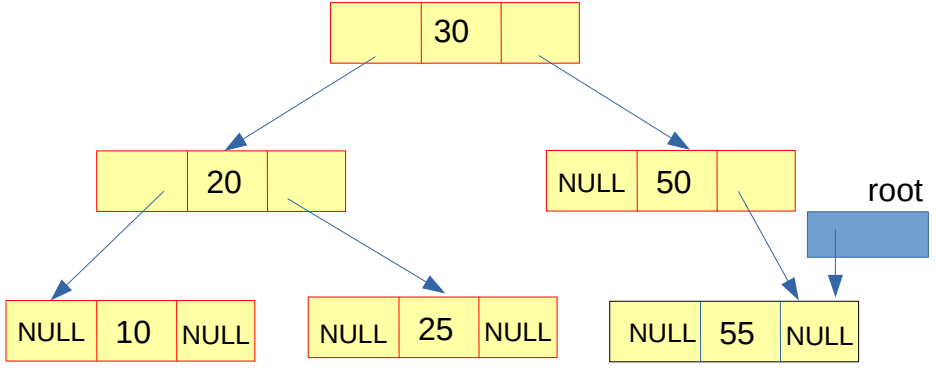
Visit Right child

EMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20 25 30 50 55




Visit Left child

Visit Parent

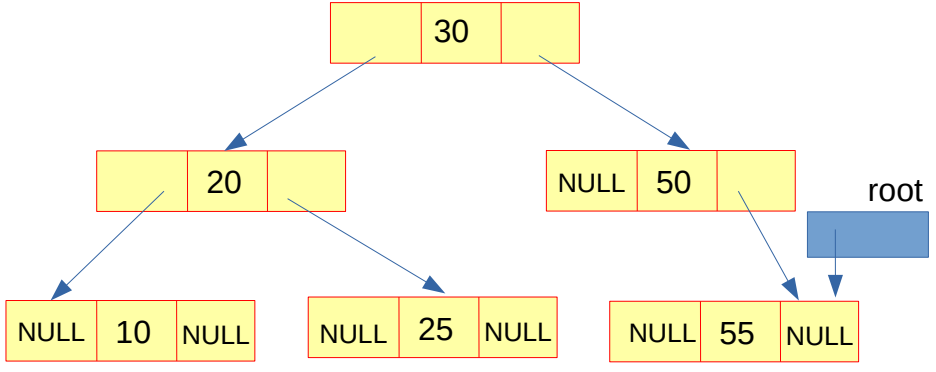
Visit Right child

ΣMERTXE

Data Structure – Binary Search Tree -Traversal

# Inorder

OUTPUT = 10 20 25 30 50 55




Visit Left child

Visit Parent

Visit Right child

EMERTXE

# Algorithm

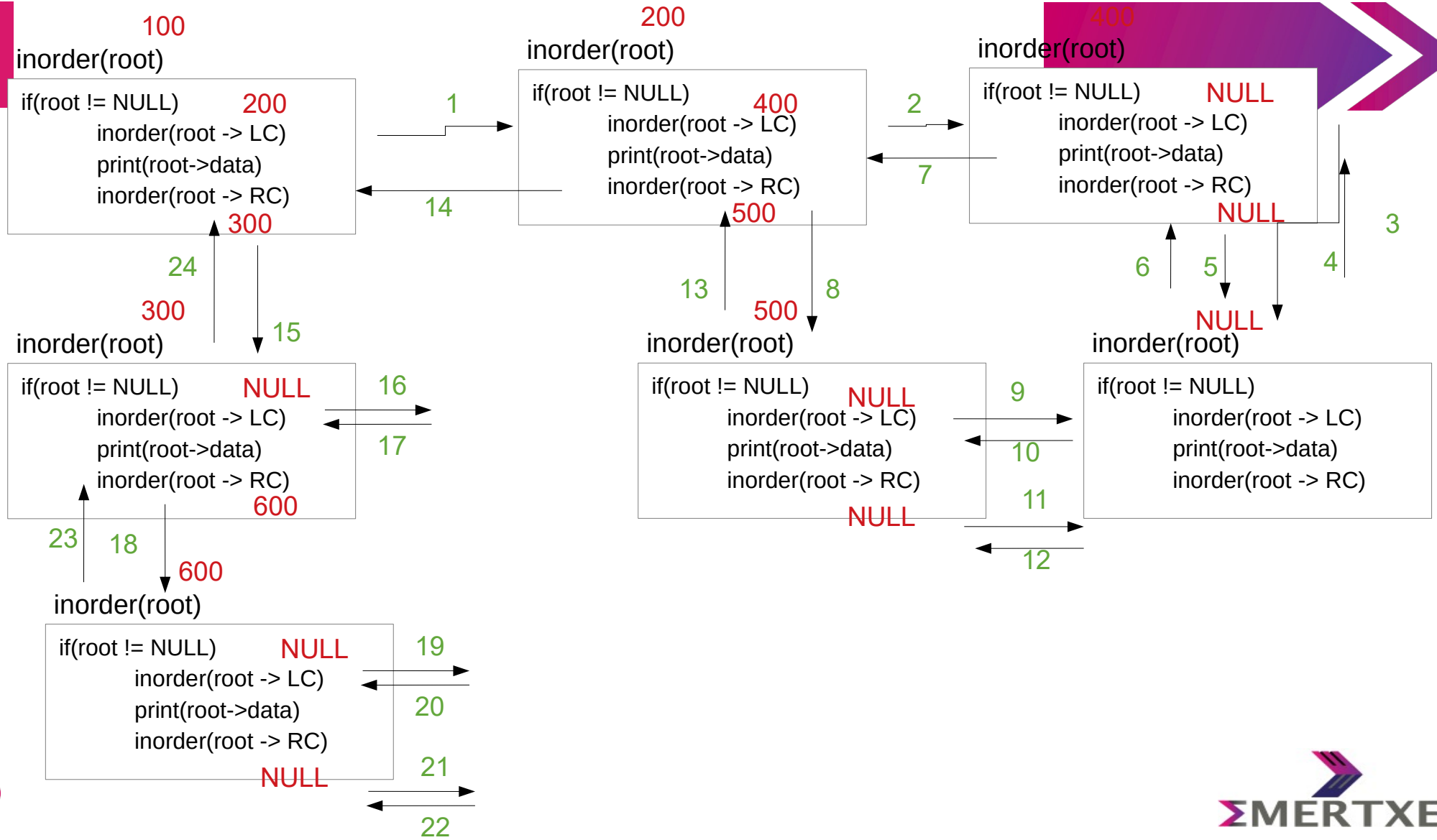
inorder(root)

```
if(root != NULL)
        inorder(root -> LC)
        print(root->data)
        inorder(root -> RC)
```

Time Complexity = O(n)

Where

‘N’ is number of node


100
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
200
300
200
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
400
500
400
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
NULL
NULL
300
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
NULL
600
500
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
NULL
NULL
600
inorder(root)
if(root != NULL)
inorder(root -> LC)
print(root->data)
inorder(root -> RC)
NULL
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

# Code - inorder(root)