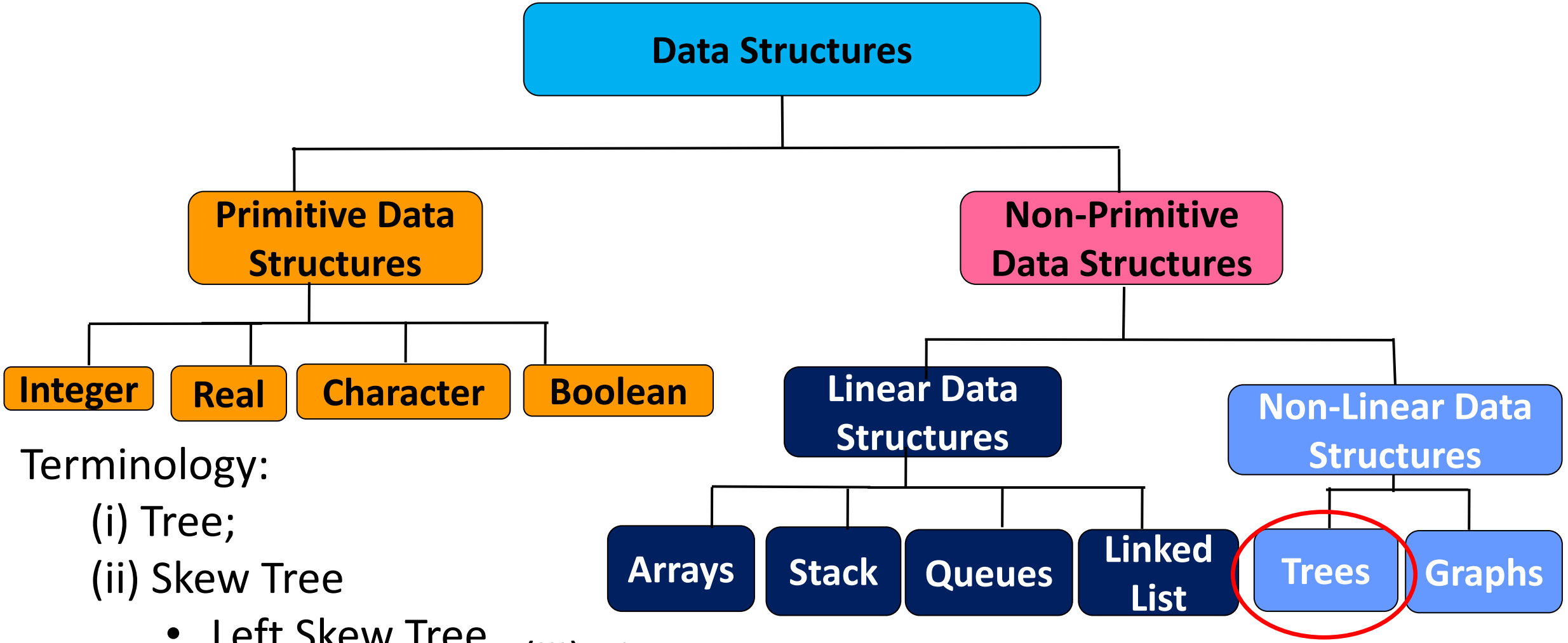
CS2x1:Data Structures and Algorithms

Koteswararao Kondepu

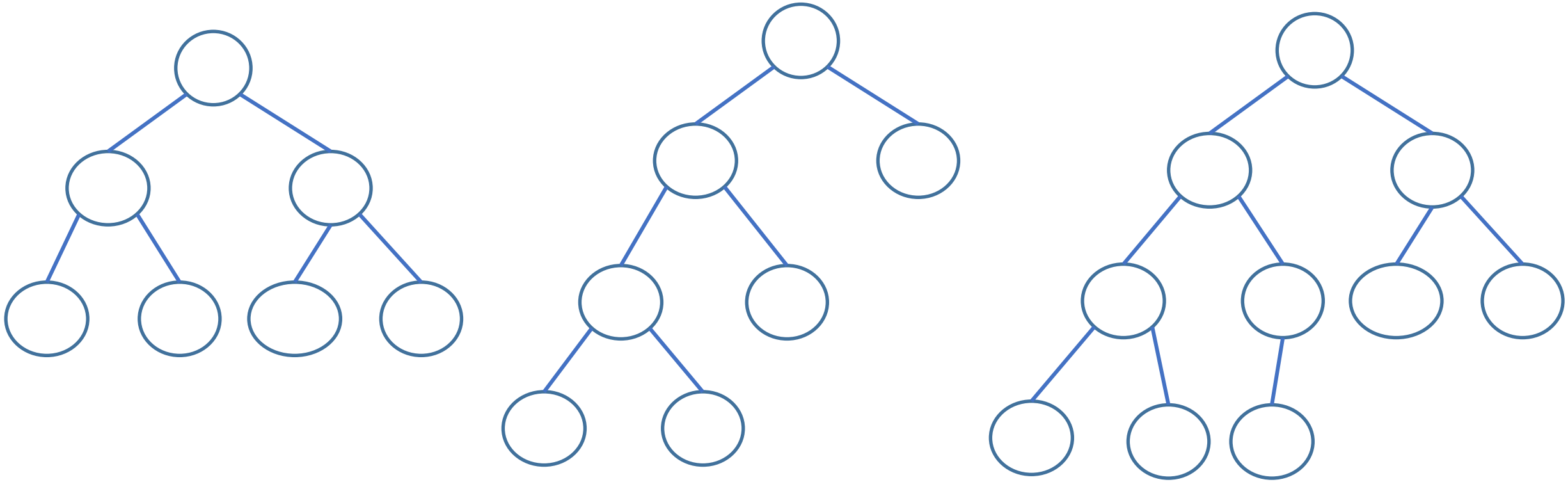
k.kondepu@iitdh.ac.in

Recap: Binary Tree



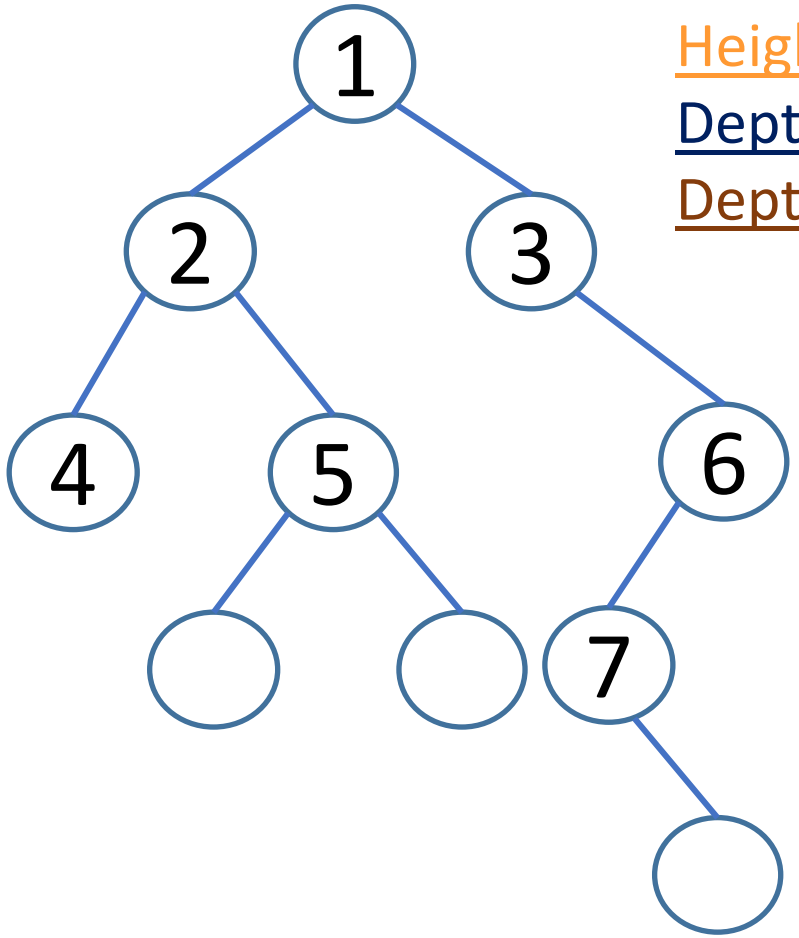
- Terminology:
  - (i) Tree;
  - (ii) Skew Tree
    - Left Skew Tree
    - Right Skew Tree
  - (iii) Binary Tree
    - Full Binary Tree
    - Perfect Binary Tree
    - Complete Binary Tree
    - Balanced Binary Tree

Exercise: Binary Tree (1)



Full Binary Tree
Perfect Binary Tree
Complete Binary Tree

Exercise: Binary Tree (2)



Height of a node: Length of the path from that node to the deepest node in the tree

Height of a tree: Maximum height among all the nodes in tree

Depth of a node: Length of the path from the root to the node

Depth of a tree: Maximum depth among all the nodes in tree

Height of node 2 →

Height of node 3 →

Height of the tree →

Depth of node 5 →

Depth of node 6 →

Depth of binary tree →

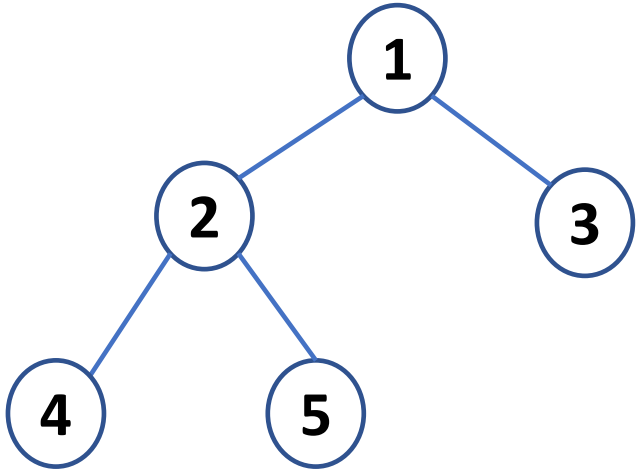
Depth of node 1 →

Binary Tree: Traversal (3)

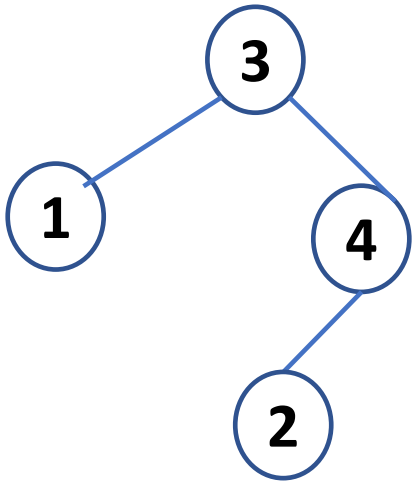
Preorder → Root, Left, Right [RT, L, R]

Inorder → Left, Root, Right [L, RT, R]

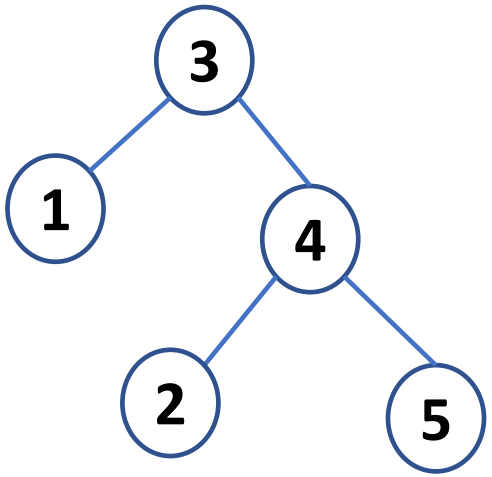
Postorder → Left, Right, Root [L, R, RT]



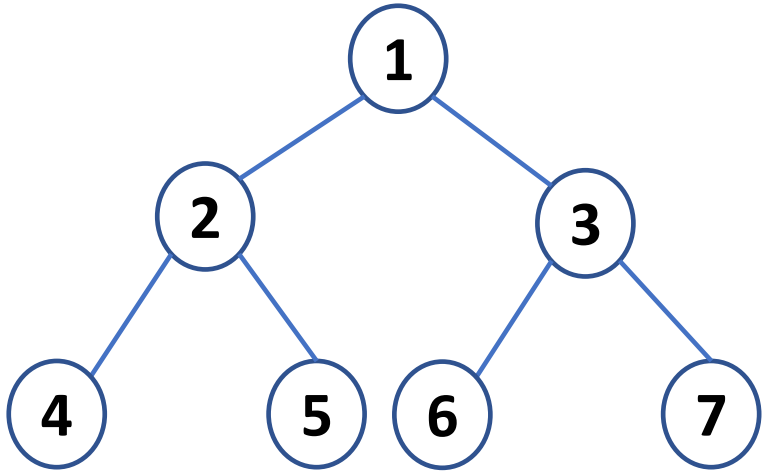
Pre-order :
In-order :
Post-order :



Pre-order:
In-order :
Post-order:



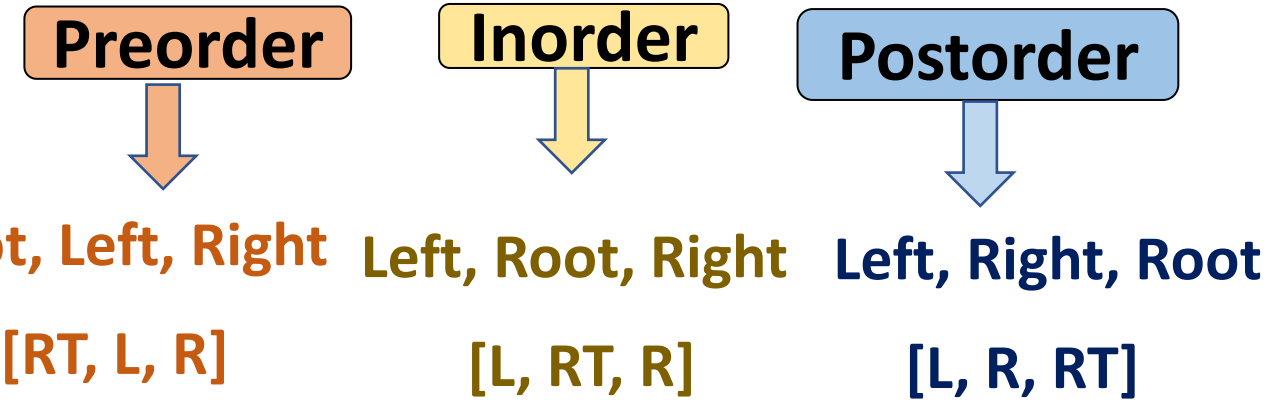
Pre-order :
In-order :
Post-order:



Pre-order :
In-order :
Post-order:

Exercise: Binary Tree Traversal (1)

Travel Policies →



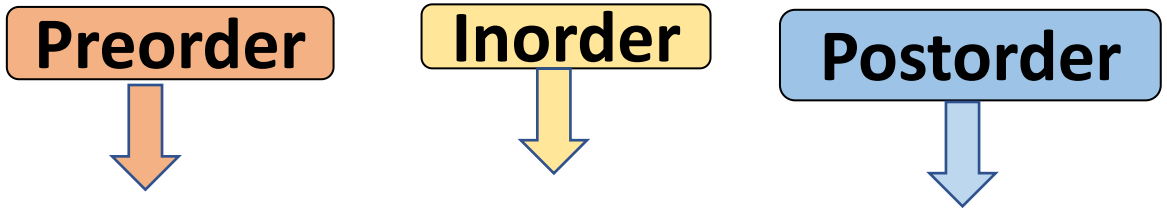
Inorder → DBEAFC
Preorder → ABDECF

What is the post-order traversal sequence of resultant tree?

Postorder →

Exercise: Binary Tree Traversal (2)

Travel Policies →



Root, Left, Right

[RT, L, R]

Left, Root, Right

[L, RT, R]

Left, Right, Root

[L, R, RT]

Inorder → 8,6,9,4,7,2,5,1,3

Postorder → 8,9,6,7,4,5,2,3,1

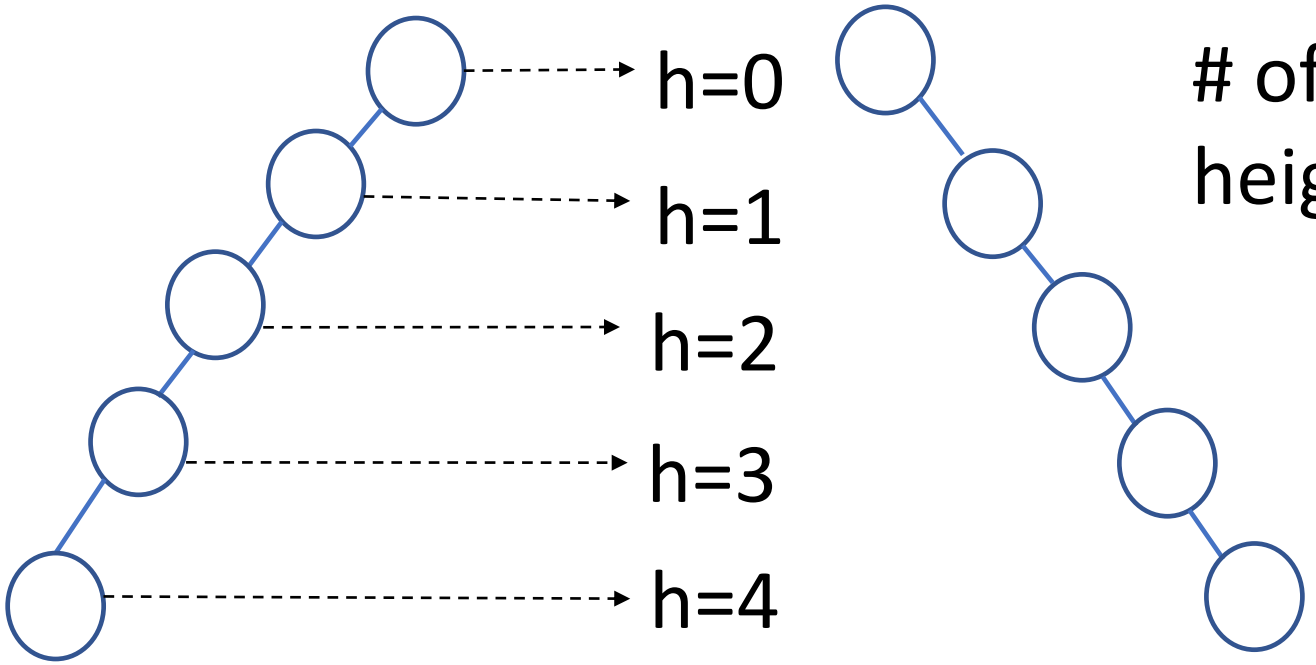
What is the pre-order traversal sequence of resultant tree?

Preorder →

Properties of Binary Tree

- Maximum height *h* of a binary tree with *n* nodes?
- Minimum height *h* of a binary tree with n nodes?
- Maximum number of external nodes in a tree of height *h*?
- Minimum number of external nodes in a tree of height *h*?
- Maximum number of internal nodes in a tree of height *h*?
- Minimum number of internal nodes in a tree of height *h*?

Maximum height h of a binary tree with n nodes?

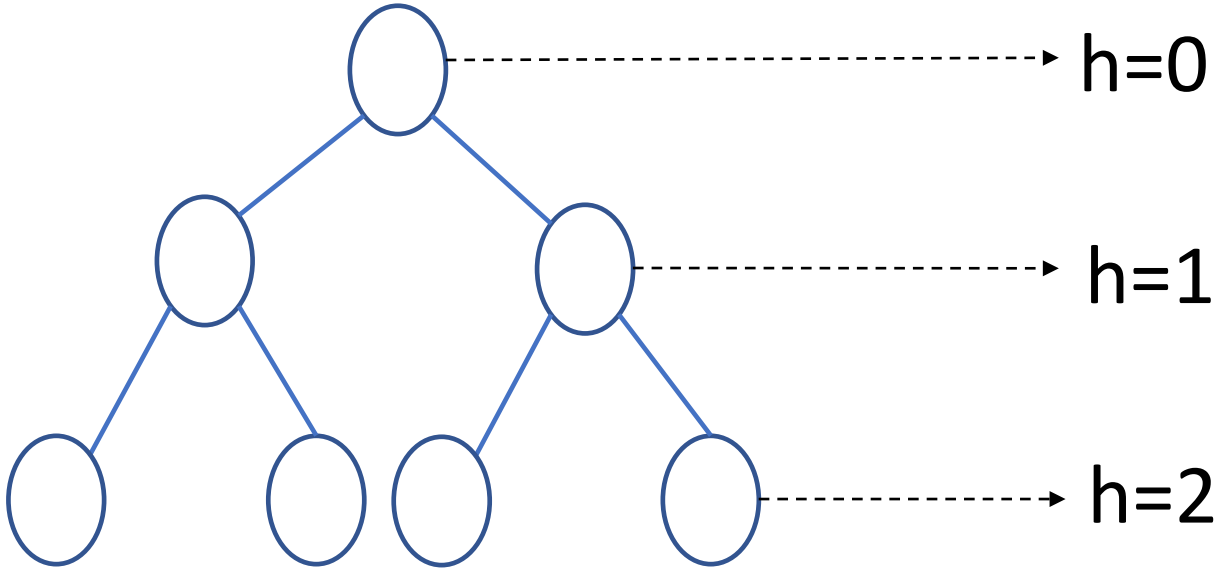


# of nodes [*n*] = 5

height of binary tree [*h*] = 4

General Case: $h \leq n-1$

Minimum height h of a binary tree with n nodes?



Perfect Binary Tree

# of nodes [*n*] = 7

height of binary tree [*h*] = 2

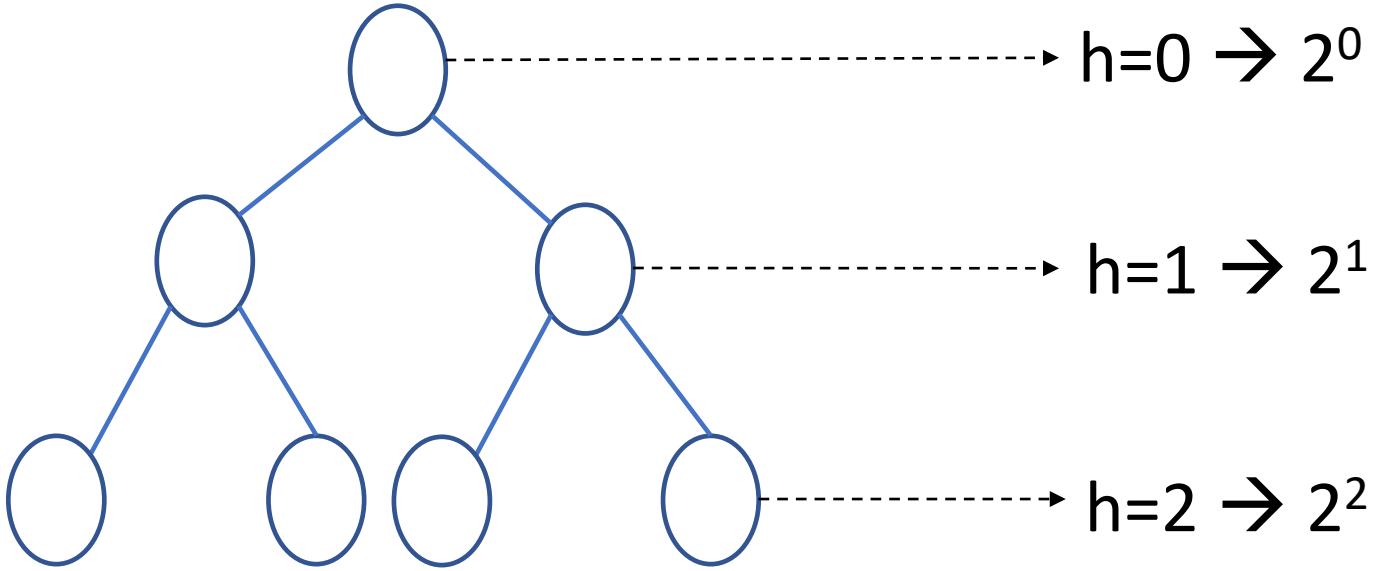
= log (n+1) – 1

= log (7+1) – 1

= 2

<u>General Case:</u> h ≥ log (n+1) -1

of nodes in the given binary tree?



Perfect Binary Tree

# of nodes (n) = 7

height of binary tree (h) = 2

= log (n+1) – 1

= log (7+1) – 1

= 2

$$2^0 + 2^1 + 2^2 = 7$$

$$= 2^{h+1} - 1$$

<u>General Case:</u> $h+1 \leq n \leq 2^{h+1} - 1$

Exercise: Binary Tree (1)

The height of a tree is the length of the longest root-to-leaf path in it. The maximum and the minimum number of nodes in a binary tree of height 5 are:

Note: the height of root h is 0

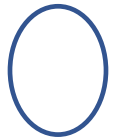
(a) 63 and 6

(b) 64 and 5

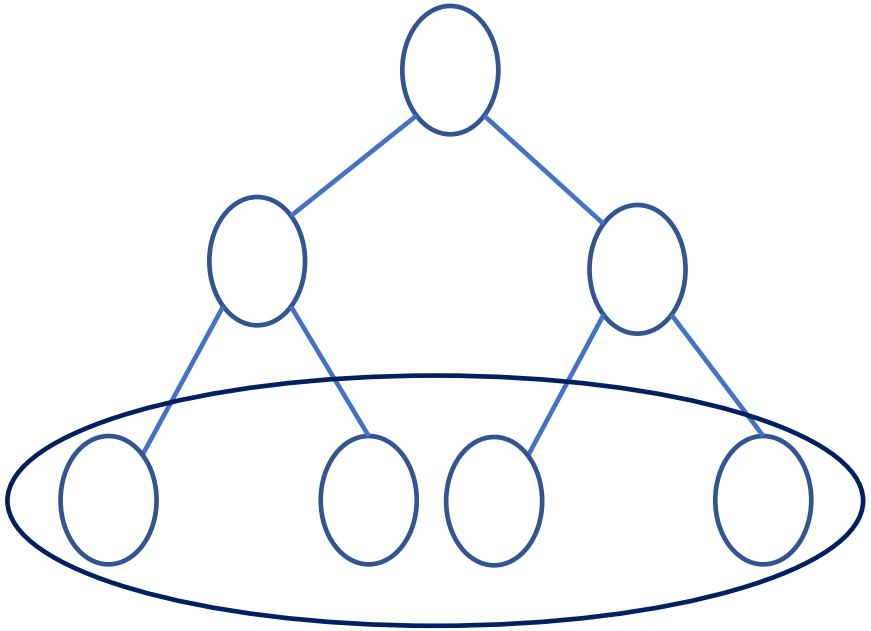
(c) 32 and 6

(d) 31 and 5

Max. and Min. number of external nodes in a tree of height *h*?



h=0→ 2^0 → 1



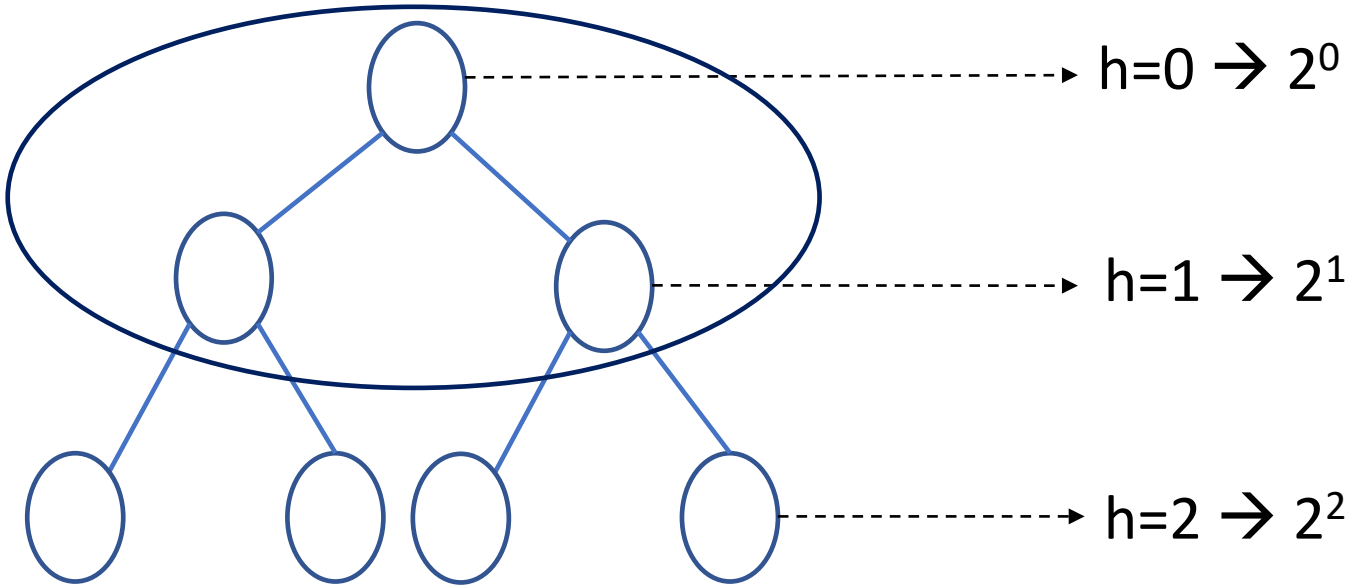
h=2 → 2^2

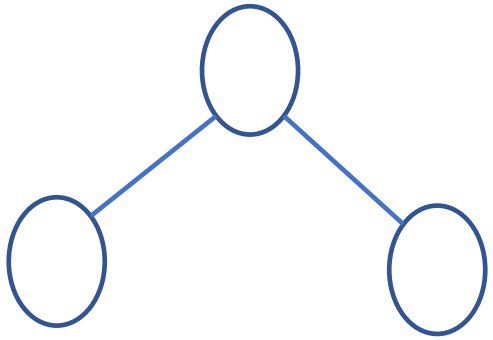
Maximum number of external nodes = 2^h

Minimum number of external nodes = 1

General case: $1 \leq n_{ext} \leq 2^h$

Max. and Min. number of internal nodes in a tree of height *h*?





h=0 → 2^0

With at least one child node

Maximum number of internal nodes = 3 = $2^h - 1$

Minimum number of internal nodes = 1

General case: $1 \leq n_{int} \leq 2^h - 1$

Exercise: Binary Tree (2)

In the binary tree, the number of internal nodes of degree 1 is 5. And, number of internal nodes of degree 2 is 10. The number of leaf nodes in the binary tree is:

(A)10

(B)11

(C)12

(D)15

Exercise: Ternary Tree (3)

In the ternary tree, the number of internal nodes of degree 2 is 1. And, number of internal nodes of degree 3 is 3. The number of leaf nodes in the ternary tree is:

(A)9

(B)10

(C)11

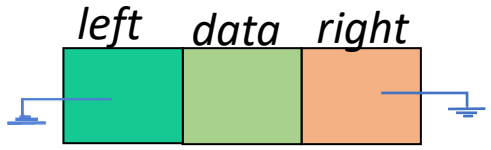
(D)8

Motivation → Binary Search Tree

- *Auxiliary Operations: Find a minimum and maximum number from the given binary tree*

```
struct btree {
    struct btree *left;
    int data;
struct btree *right; }
```

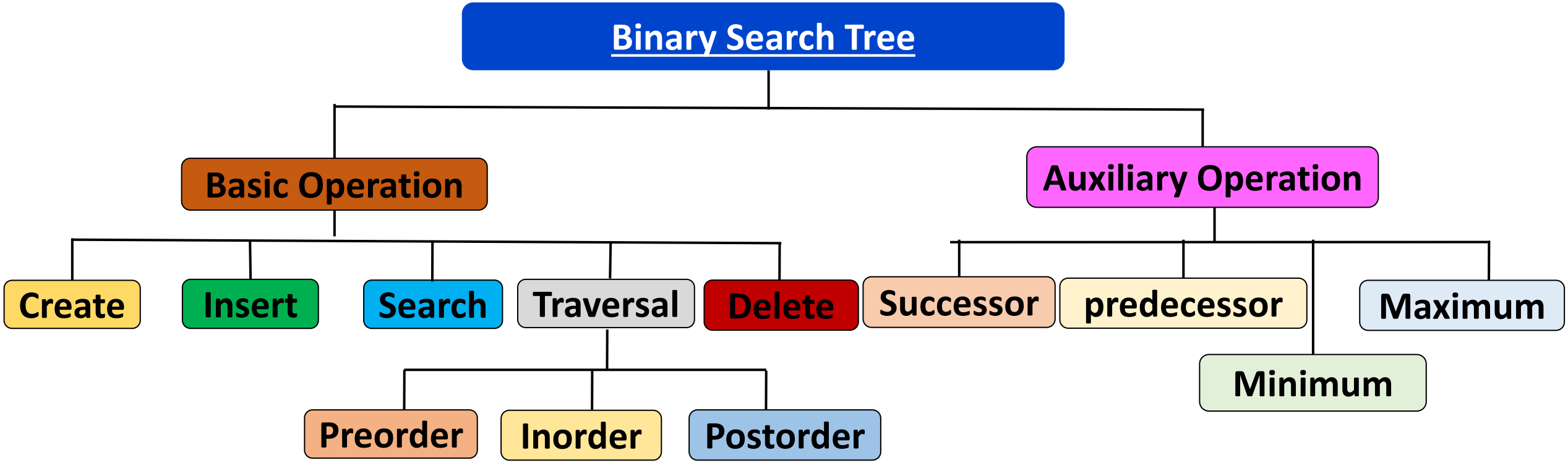
```
int FindMax(struct btree *root){
    int rval, leftv, rightv, max;
    struct btree *fmax;
    fmax=root;
    if (fmax!=NULL){
        rval=fmax->data;
        leftv=FindMax(fmax->left);
        rightv=FindMax(fmax->right);
        max= (leftv > rightv)?leftv:rightv;
        max= (rval > max)?rval:max;
    }
   return max;
}
```



btree → binary tree

Time Complexity O (*n*) n: # of nodes in a binary tree

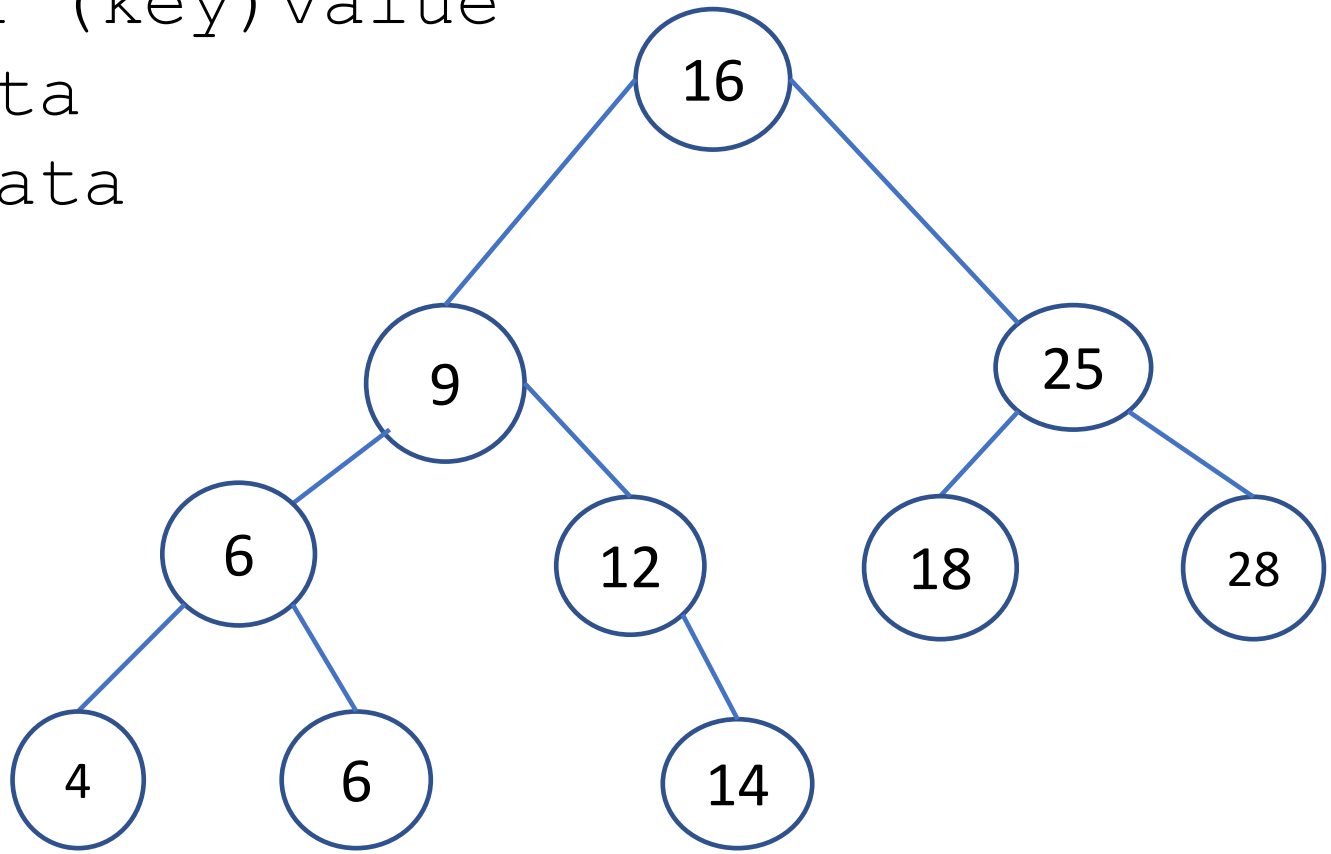
Binary Search Tree



Binary Search Tree: Properties

```
struct BSTree {
        struct BSTree *left;
        int data;
    struct BSTree *right;}
```

- Each node contains a data (key)value
- Left Subtree < root → data
- Right Subtree > root → data



thank you!

email:
k.kondepu@iitdh.ac.in