



CSE 247

Data Structures

Outline

- Perform the following operation in the binary tree and understand their big-Oh.
 - Calculate the height of a binary tree
 - Maximum number of nodes in a binary tree
 - Minimum possible height of a binary tree with N nodes
 - Count the total number of leaves nodes in a binary tree
- Types of binary trees
 - Complete binary tree
 - Full binary tree
 - Perfect binary tree
- Other Applications
 - Huffman codes, a compression technique
 - Heap tree to build a priority queue

Calculate height

Algorithm 1: Algorithm for calculating the height of a binary tree

Data: *root*: root node of the binary tree

Result: Height of the binary tree

Procedure **BTHight**(*root*)

if *root* == *NULL* **then**

 | *return* 0;

end

$leftTHight = BTHight(root \rightarrow left);$

$rightTHight = BTHight(root \rightarrow right);$

return $Max(leftTHight, rightTHight) + 1;$

Calculate height of a binary tree

Implement recursive function to calculate height of binary tree.

```
public int heightCount(node temp) {  
    int heightLeft;  
    int heightRight;  
    if(temp != NULL){  
        heightLeft=heightCount(temp.left);  
        heightRight=heightCount(temp.right);  
        return Math.max(heightLeft,heightRight)+1;  
    }  
}
```

Recursive function to Count leaf nodes

```
public int LeafCount(node temp) {
```

base case :

```
if(temp == NULL)    return 0;
```

```
if(temp.left == NULL && temp.right==NULL) return 1;
```

Recursive call:

```
return LeafCount(temp.left)+ LeafCount(temp.right);
```

```
}
```

Binary trees

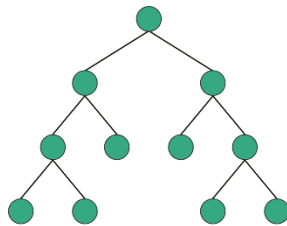
- ***What is the maximum number of nodes in a Binary tree with height h ?***
- Answer
 - when the Height of a tree with a single node is considered as 1, the maximum number of nodes in a binary tree of height h is
$$1 + 2 + 4 + \dots + 2^{h-1} \text{ and}$$
the sum of this series is $2^h - 1$.
 - when the Height of a tree with a single node is considered as 0. In this convention, the above formula becomes $2^{h+1} - 1$

Operations on trees

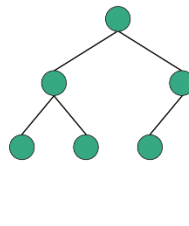
- ***What is the minimum possible height of the binary tree with n nodes?***
- Answer
 - In a Binary Tree with N nodes, minimum possible height or the minimum number of levels is $\text{Log}_2(N+1)$.
 - If we consider the convention where the height of a leaf node is considered as 0, then above formula for minimum possible height becomes $\text{Log}_2(N+1) - 1$

Five Types of Binary Trees

- Complete binary tree: A complete binary tree of height h has all possible nodes through level $h-1$, and some nodes on depth h exist left to right with no gaps.
- Full binary tree: A tree in which every node other than the leaves has two children
- Perfect binary tree: A *full binary tree* in which all *leaves* are at the same *level*



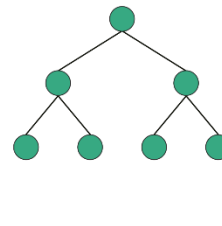
Full



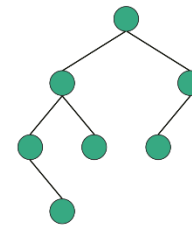
Complete



Degenerate



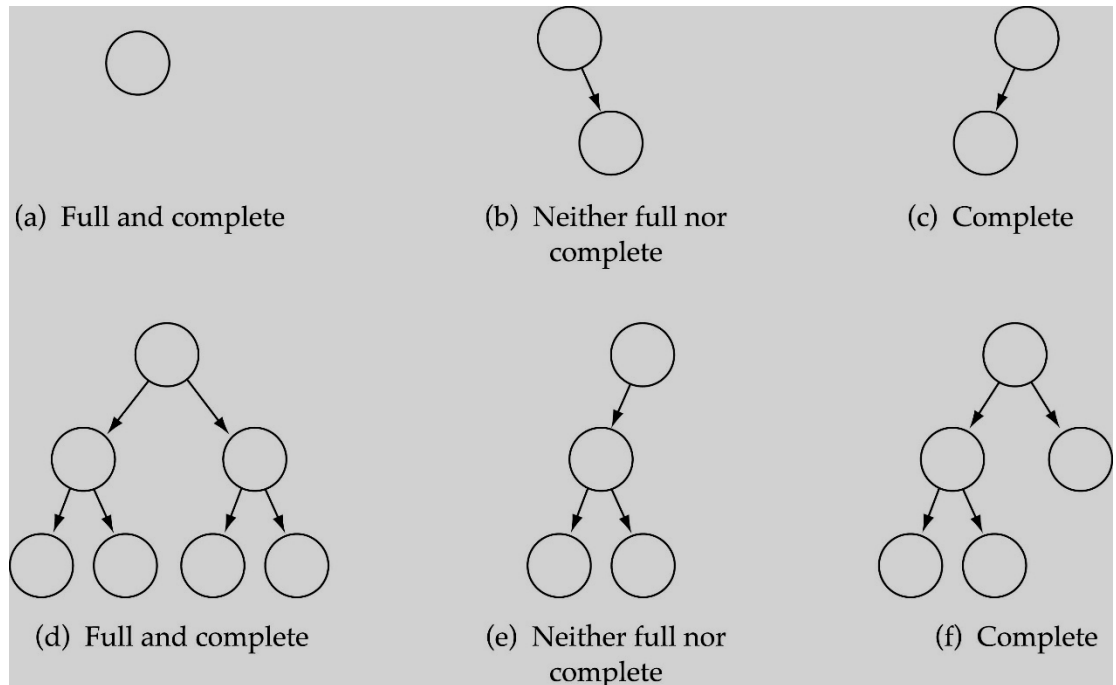
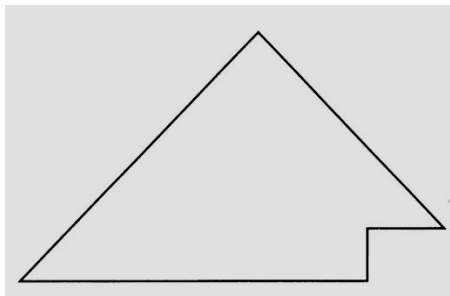
Perfect



Balanced

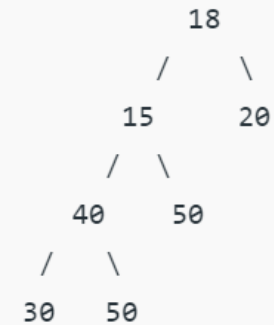
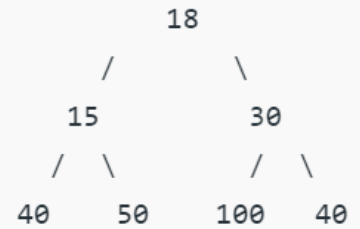
Complete Binary Tree

- A binary tree that is either full or full through the next-to-last level
- The last level is full from left to right (i.e., leaves are as far to the left as possible)
- Height=

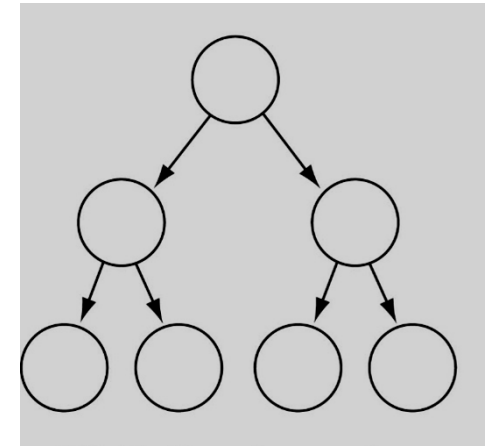


Full Binary Tree

- Every non-leaf node has two children



Perfect binary tree



- Every non-leaf node has two children
- All the leaves are on the same level
- A Perfect Binary Tree of height h has $2^h - 1$ nodes if height start from 1.
- A perfect binary tree of height h has $2^{h+1} - 1$ node if height start from 0.
- A perfect binary tree with n nodes has height $\log(n + 1) - 1 = O(\log(n))$.
- A perfect binary tree of height h has 2^h leaf nodes if height start from 0.
- A perfect binary tree of height h has $2^h - 1$ leaf nodes if height start from 1.