Trees

# **Binary Trees:**

**1) The maximum number of nodes at level ‘l’ of a binary tree is 2l**.   
Here level is the number of nodes on the path from the root to the node (including root and node). Level of the root is 0.   
This can be proved by induction.   
For root, l = 0, number of nodes = 20 = 1

**2) The** **Maximum number of nodes in a binary tree of height ‘h’ is 2h – 1**.   
Here the height of a tree is the maximum number of nodes on the root to leaf path. Height of a tree with a single node is considered as 1.   
maximum number of nodes in a binary tree of height h is 1 + 2 + 4 + .. + 2h-1. This is a simple geometric series with h terms and sum of this series is 2h– 1.   
In some books, the height of the root is considered as 0. In this convention, the above formula becomes 2h+1 – 1

**Depth of BT=max[no. of levels]**

**Height of BT=1+max(height)**

**3) In a Binary Tree with N nodes, minimum possible height or the minimum number of levels is Log2(N+1).**  
There should be at least one element on each level, so the height cannot be more than N. A binary tree of height ‘h’ can have maximum 2h – 1 nodes (previous property). So the number of nodes will be less than or equal to this maximum value.

N <= 2h - 1

2h >= N+1

log2(2h) >= log2(N+1) (Taking log both sides)

hlog22 >= log2(N+1) (h is an integer)

h >= | log2(N+1) |

So the minimum height possible is | log2(N+1) |

**4) A Binary Tree with L leaves has at least | Log2L |+ 1   levels.**   
A Binary tree has the maximum number of leaves (and a minimum number of levels) when all levels are fully filled. Let all leaves be at level l, then below is true for the number of leaves L.

L <= 2l-1 [From Point 1]

l = | Log2L | + 1

where l is the minimum number of levels.

**5) In Binary tree where every node has 0 or 2 children, the** **number of leaf nodes is always one more than nodes with two children**.

L = T + 1

Where L = Number of leaf nodes

T = Number of internal nodes with two children

Proof:

No. of leaf nodes (L) i.e. total elements present at the bottom of tree =

2h-1 (h is height of tree)

No. of internal nodes = {total no. of nodes} - {leaf nodes} =

{ 2h - 1 } - {2h-1} = 2h-1 (2-1) - 1 = 2h-1 - 1

So , L = 2h-1

T = 2h-1 - 1

Therefore L = T + 1

Hence proved

**6) In a non empty binary tree, if n is the total number of nodes and e is the total number of edges, then e = n-1**

Every node in a binary tree has exactly one parent with the exception of root node. So if n is the total  
number of nodes then n-1 nodes have exactly one parent. There is only one edge between any child and its  
parent. So the total number of edges is n-1.