***1) The maximum number of nodes at level ‘l’ of a binary tree is 2l-1***.  
Here level is number of nodes on path from root to the node (including root and node). Level of root is 1.  
This can be proved by induction.  
For root, l = 1, number of nodes = 21-1 = 1  
Assume that maximum number of nodes on level l is 2l-1  
Since in Binary tree every node has at most 2 children, next level would have twice nodes, i.e. 2 \* 2l-1

***2) Maximum number of nodes in a binary tree of height ‘h’ is 2h – 1***.  
Here height of a tree is maximum number of nodes on root to leaf path. Height of a tree with single node is considered as 1.  
This result can be derived from point 2 above. A tree has maximum nodes if all levels have maximum nodes. So 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, height of the root is considered as 0. In this convention, the above formula becomes 2h+1 – 1

***3) In a Binary Tree with N nodes, minimum possible height or minimum number of levels is  ? Log2(N+1) ?***  
This can be directly derived from point 2 above. If we consider the convention where height of a leaf node is considered as 0, then above formula for minimum possible height becomes   ? Log2(N+1) ? – 1

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

Binary Tree | Set 3 (Types of Binary Tree)

We have discussed [Introduction to Binary Tree in set 1](http://quiz.geeksforgeeks.org/binary-tree-set-1-introduction/) and [Properties of Binary Tree in Set 2](http://quiz.geeksforgeeks.org/binary-tree-set-2-properties/). In this post, common types of binary is discussed.

Following are common types of Binary Trees.

**Full Binary Tree** A Binary Tree is full if every node has 0 or 2 children. Following are examples of a full binary tree. We can also say a full binary tree is a binary tree in which all nodes except leaves have two children.

18

/ \

15 30

/ \ / \

40 50 100 40

18

/ \

15 20

/ \

40 50

/ \

30 50

18

/ \

40 30

/ \

100 40

***In a Full Binary, number of leaf nodes is number of internal nodes plus 1***  
       L = I + 1  
Where L = Number of leaf nodes, I = Number of internal nodes  
See [Handshaking Lemma and Tree](https://www.geeksforgeeks.org/handshaking-lemma-and-interesting-tree-properties/) for proof.

**Complete Binary Tree:** A Binary Tree is complete Binary Tree if all levels are completely filled except possibly the last level and the last level has all keys as left as possible

Following are examples of Complete Binary Trees

18

/ \

15 30

/ \ / \

40 50 100 40

18

/ \

15 30

/ \ / \

40 50 100 40

/ \ /

8 7 9

Practical example of Complete Binary Tree is [Binary Heap](http://quiz.geeksforgeeks.org/binary-heap/).

**Perfect Binary Tree** A Binary tree is Perfect Binary Tree in which all internal nodes have two children and all leaves are at the same level.  
Following are examples of Perfect Binary Trees.

18

/ \

15 30

/ \ / \

40 50 100 40

18

/ \

15 30

A Perfect Binary Tree of height h (where height is the number of nodes on the path from the root to leaf) has 2h – 1 node.

Example of a Perfect binary tree is ancestors in the family. Keep a person at root, parents as children, parents of parents as their children.

**Balanced Binary Tree**  
A binary tree is balanced if the height of the tree is O(Log n) where n is the number of nodes. For Example, AVL tree maintains O(Log n) height by making sure that the difference between heights of left and right subtrees is atmost 1. Red-Black trees maintain O(Log n) height by making sure that the number of Black nodes on every root to leaf paths are same and there are no adjacent red nodes. Balanced Binary Search trees are performance wise good as they provide O(log n) time for search, insert and delete.

**A degenerate (or pathologyical) tree**A Tree where every internal node has one child. Such trees are performance-wise same as linked list.

10

/

20

\

30

\

40

Enumeration of Binary Trees

A Binary Tree is labeled if every node is assigned a label and a Binary Tree is unlabeled if nodes are not assigned any label.

Below two are considered same unlabeled trees

o o

/ \ / \

o o o o

Below two are considered different labeled trees

A C

/ \ / \

B C A B

**How many different Unlabeled Binary Trees can be there with n nodes?**

For n = 1, there is only one tree

o

For n = 2, there are two trees

o o

/ \

o o

For n = 3, there are five trees

o o o o o

/ \ / \ / \

o o o o o o

/ \ \ /

o o o o

The idea is to consider all possible pair of counts for nodes in left and right subtrees and multiply the counts for a particular pair. Finally add results of all pairs.

For example, let T(n) be count for n nodes.

T(0) = 1 [There is only 1 empty tree]

T(1) = 1

T(2) = 2

T(3) = T(0)\*T(2) + T(1)\*T(1) + T(2)\*T(0) = 1\*2 + 1\*1 + 2\*1 = 5

T(4) = T(0)\*T(3) + T(1)\*T(2) + T(2)\*T(1) + T(3)\*T(0)

= 1\*5 + 1\*2 + 2\*1 + 5\*1

= 14

The above pattern basically represents [n’th Catalan Numbers](https://www.geeksforgeeks.org/program-nth-catalan-number/). First few catalan numbers are 1 1 2 5 14 42 132 429 1430 4862,…  
  
Here,  
T(i-1) represents number of nodes on the left-sub-tree  
T(n−i-1) represents number of nodes on the right-sub-tree

n’th Catalan Number can also be evaluated using direct formula.

T(n) = (2n)! / (n+1)!n!