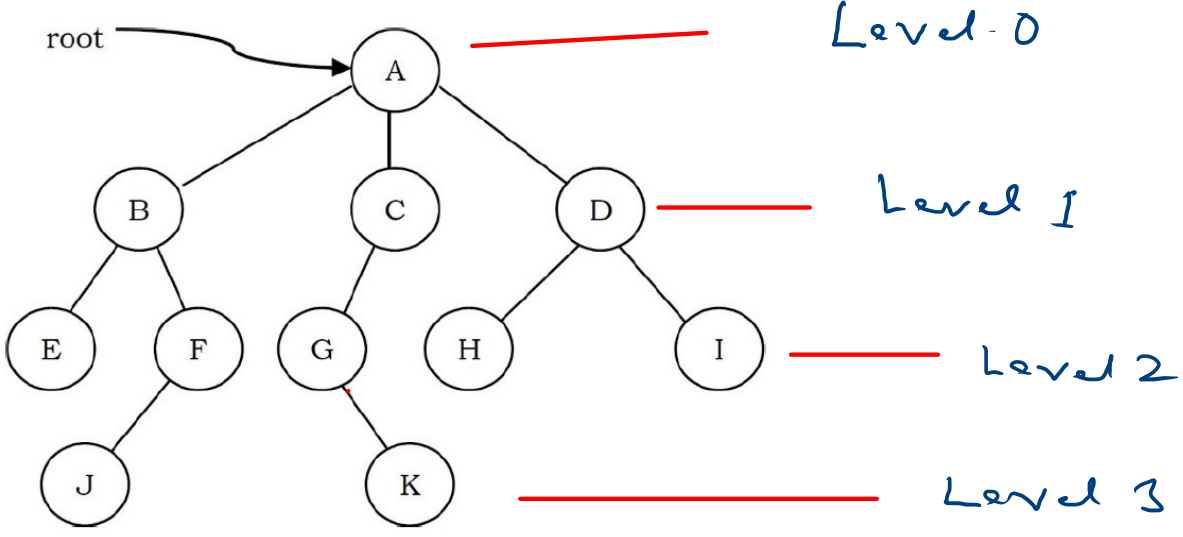


DEPTH : The depth of a node is the length of the path from the root to the node (depth of G is 2, A – C – G).

HEIGHT: The height of a node is the length of the path from that node to the deepest node.
The height of a tree is the length of the path from the root to the deepest node in the tree.
A (rooted) tree with only one node (the root) has a height of zero. In the previous
example, the height of B is 2 (B – F – J).

Depth and height of a tree returns the same value. But for individual nodes we may get different results.

SIZE : The size of a node is the number of descendants it has including itself (the size of the subtree C is 3).

SKEW TREE:

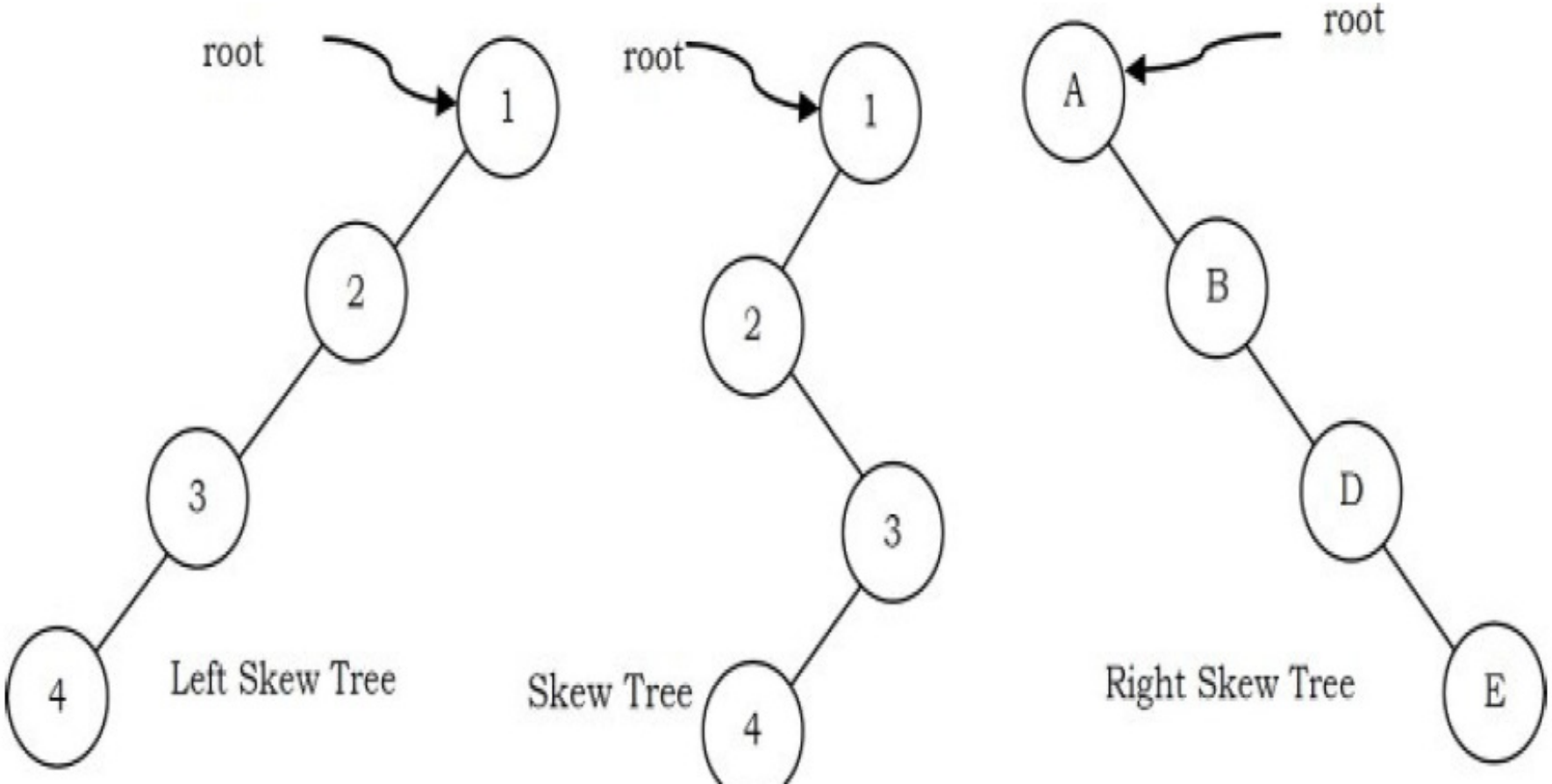
If every node in a tree has only one child (except leaf nodes) then we call such trees skew trees.

LEFT SKEW TREE:

If every node has only left child then we call them left skew trees.

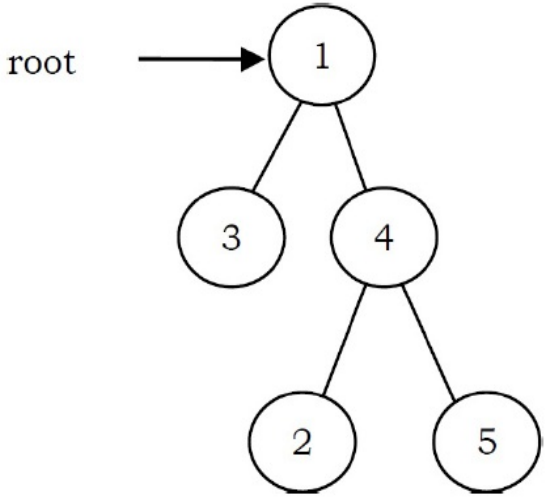
RIGHT SKEW TREE:

If every node has only right child then we call them right skew trees.

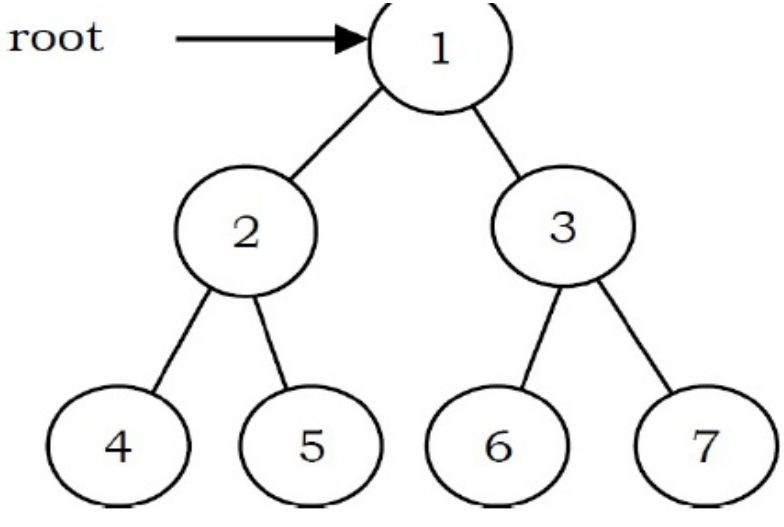


Types of Binary Trees

Strict Binary Tree: A binary tree is called strict binary tree if each non-leaf node has exactly two children. Strict binary tree can be unbalanced.



Full Binary Tree: A binary tree is called full binary tree if each node has exactly two children and all leaf nodes are at the same level.
Full binary tree is a balanced Strict binary tree.



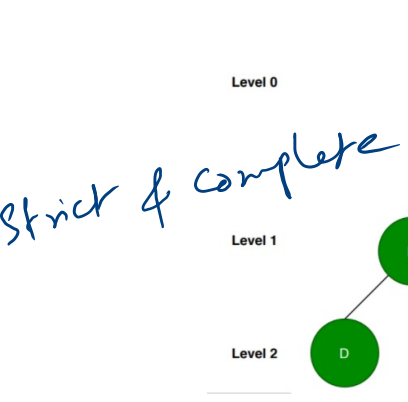
COMPLETE BINARY TREE:

In a complete binary tree, all the levels of a tree are filled entirely except the last level.

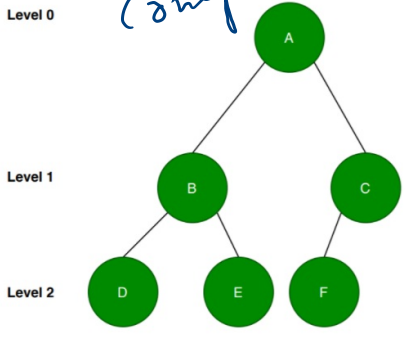
In the last level, nodes might or might not be filled fully. Also, let's note that all the nodes should be filled from the left. Since we fill the nodes from the left, so it is not allowed for a node to have a right child node without having a left child node.

It is absolutely mandatory to conatin 2^L nodes at each level except the last level.

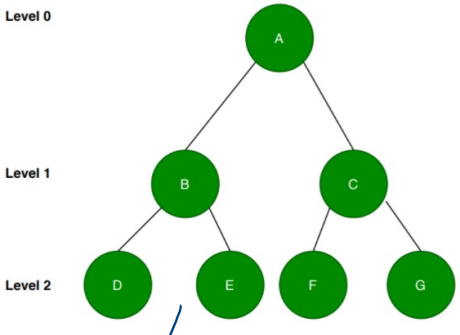
Complete binary tree can be :
Full binary tree
or Strict binary tree
or neither Full binary tree nor Strict binary tree



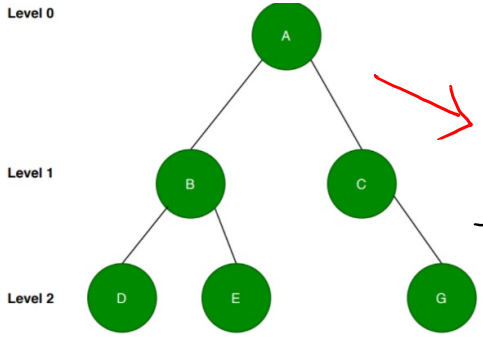
Strict & complete



complete only



strict & full & complete

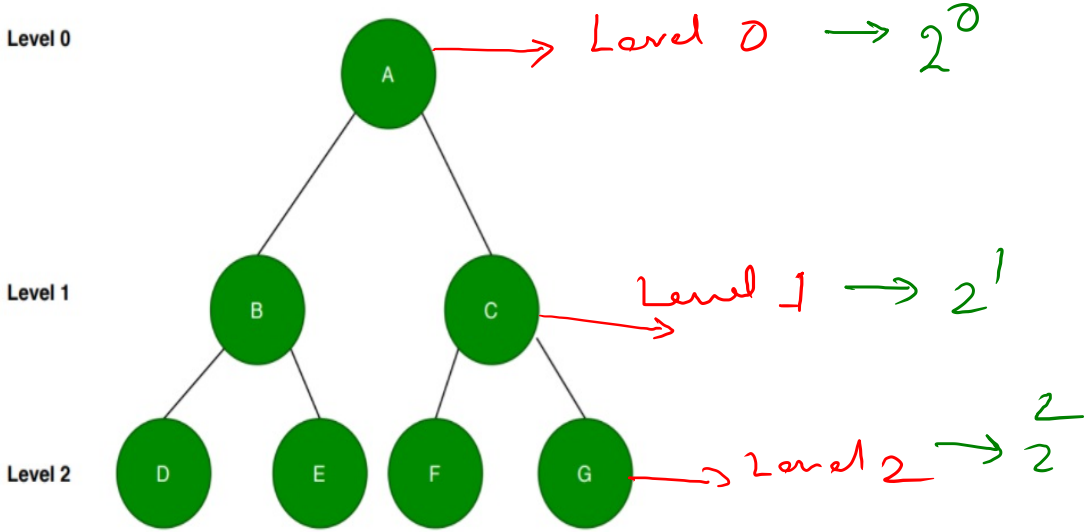


not a strict
not a full

→ Even not a complete because right child has been added before left

not a left sequential

Calculation of nodes for Full binary Tree



Level 0 → 2^0

Level 1 → 2^1

Level 2 → 2^2

Total nodes $= [2^0 + 2^1 + 2^2 + \dots 2^L]$

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

eg. If 4 levels then total nodes $= 2^{4+1} - 1$

$= 2^5 - 1$

$= 31$