# Types of Binary Trees

In the previous lecture, we learned about a special variety of trees, binary trees. We saw key characteristics of a binary tree that makes it different from other trees. Today, we'll see more about trees, binary trees, and different types of binary trees.

Earlier we learned about trees, and how it offers a non-linear representation of data. Most of the data structures we discussed in this playlist had represented data in a linear fashion, be it arrays, queues, stacks, or linked lists. Trees are ideally used to represent hierarchical data. Although I should have said most of this in the first video, few things actually become clearer as time goes on. So, trees are useful in representing an organizational hierarchy. Refer to the illustration below.
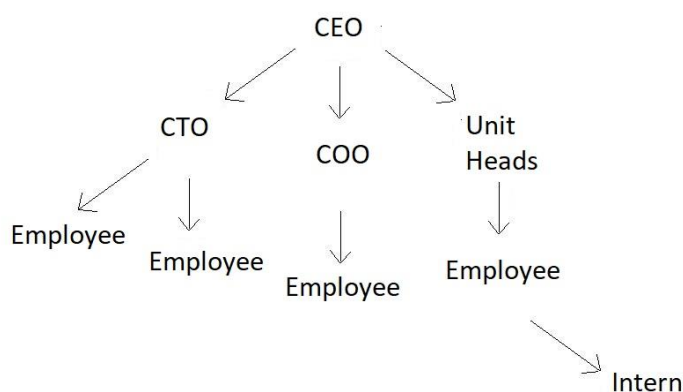

Another good example would be our file manager in desktops. There is a C Drive, it contains folders Windows, Program Files, etc. which further contain folders. So, they are hierarchically represented using a tree.

1. **Full or Strict Binary trees:**

Binary trees as we said earlier have a degree of 2 or less than 2. But a strict binary tree is a binary tree having all of its nodes with a degree of 2 or 0. That is each of its nodes either have 2 children or is a leaf node. Few simple examples follow:
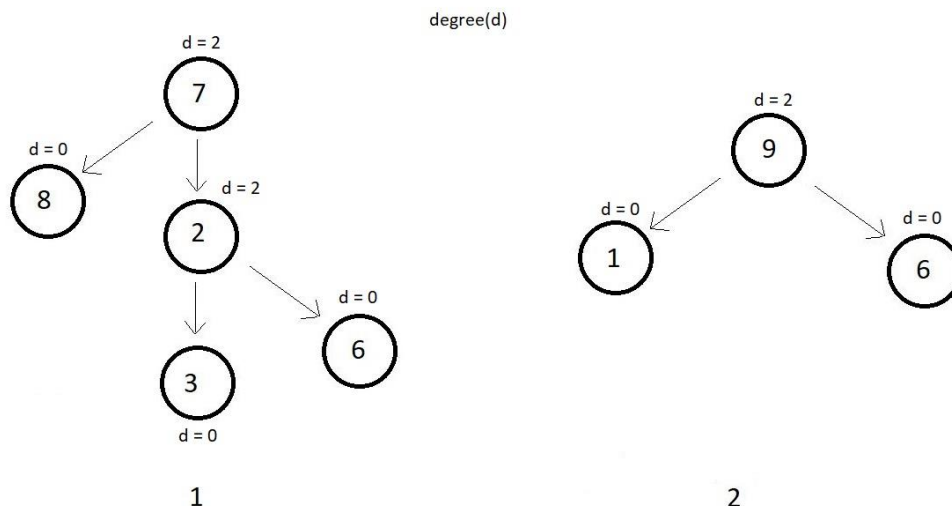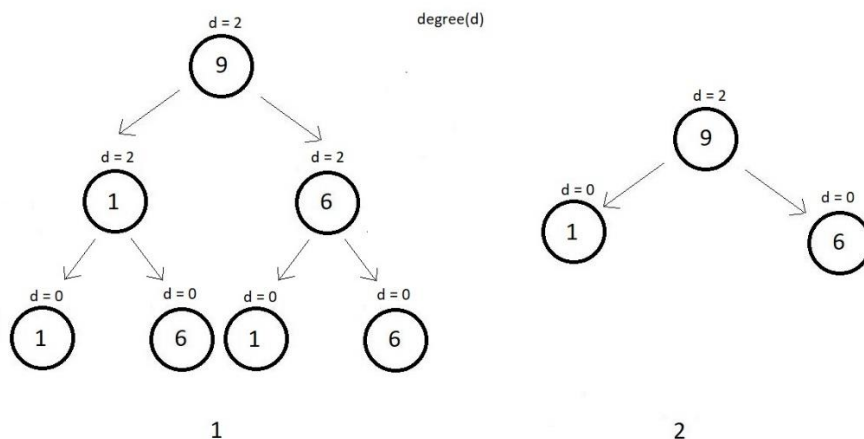
Below illustrated is a binary which is not a strict or full binary tree because the colored node has a degree of just 1.

degree(d)

d = 2
9

d = 2
1

d = 1
6

d = 0
1

d = 0
6

d = 0
1

1
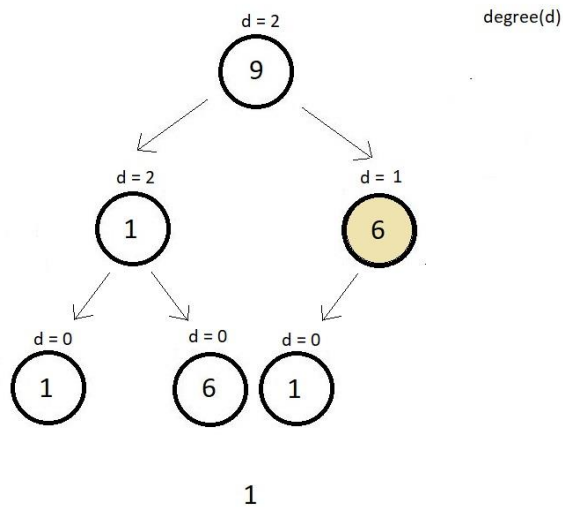
2. **Perfect Binary Tree:**

A perfect binary tree has all its internal nodes with degree strictly 2 and has all its leaf nodes on the same level. A perfect binary tree as the name suggests appears exactly perfect. Few examples follow:

degree(d)

d = 2
9

d = 2
1

d = 2
6

d = 0
1

d = 0
6

d = 0
1

d = 0
6

1

d = 2
9

d = 0
1

d = 0
6

2

Every leaf node above in both the examples are on the same level and all the internal nodes have a degree 2. Below illustrated is a binary which is not a perfect binary tree because the colored node is an internal node and has a degree of just 1, although each leaf node is on the same level.

degree(d)

d = 2
9

d = 2
1

d = 1
6

d = 0
1

d = 0
6

d = 0
6

1
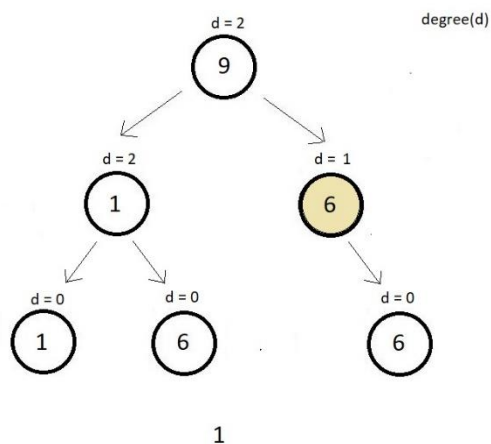
3. Complete Binary Tree:

A complete binary tree has all its levels completely filled except possibly the last level. And if the last level is not completely filled then the last level's keys must be all left-aligned. It must have sounded tough to figure out. But the illustrations below would clear all your confusion.

1   2
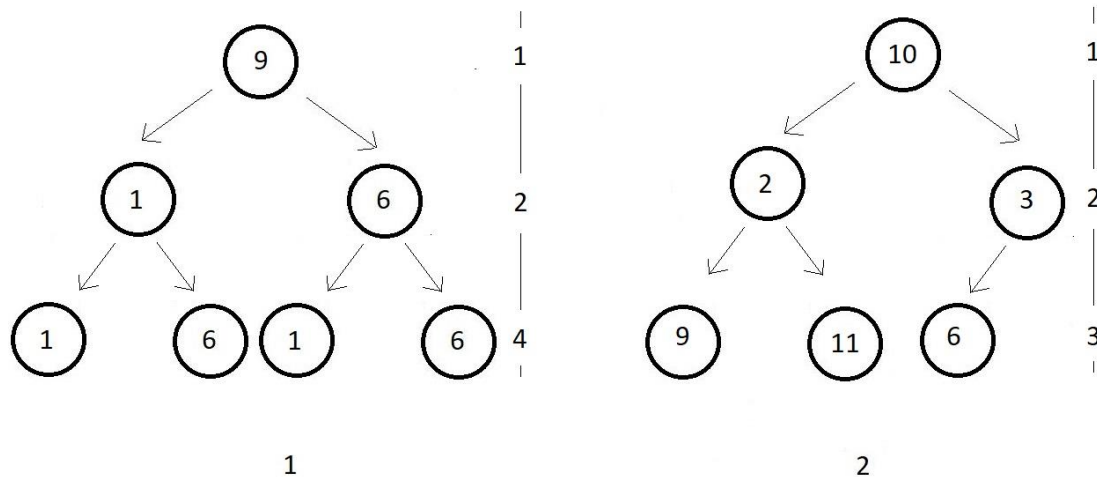
As indicated in figure 1, all levels are completely filled, so nothing further needs to be done. It is a complete binary tree. In figure 2, all nodes are completely filled except the last level which has just 3 keys. It is nonetheless a complete binary tree because all keys are left-aligned. Below illustrated are some non-complete binary trees.

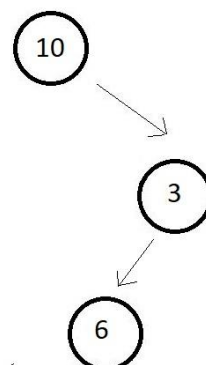1   2
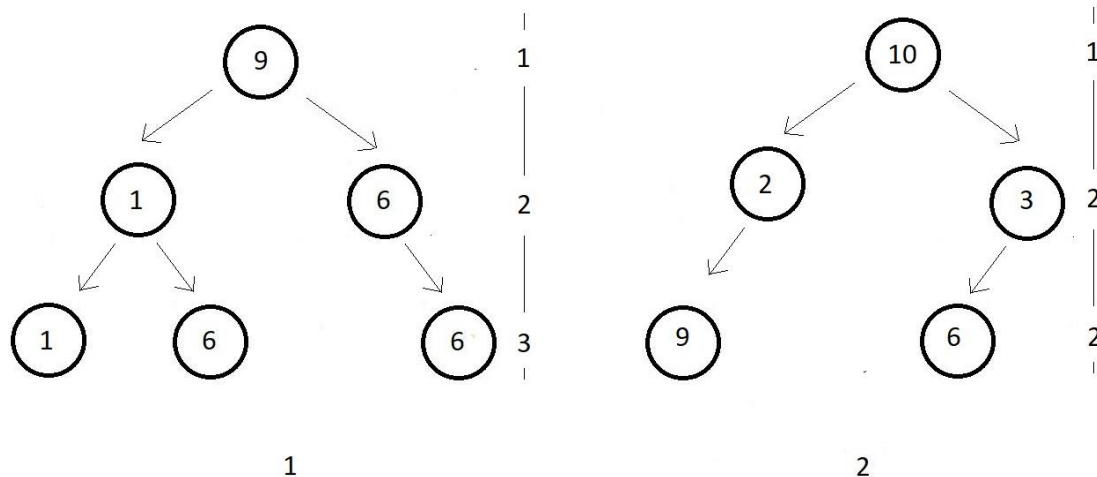
In both the figures, the last level is not complete. And hence we check if all the nodes are aligned to the left. But they aren't in both cases. And hence both of them are not complete.

4. Degenerate tree:

The easiest of all, degenerate trees are binary trees where every parent node has just one child and that can be either to its left or right. Few of its examples:
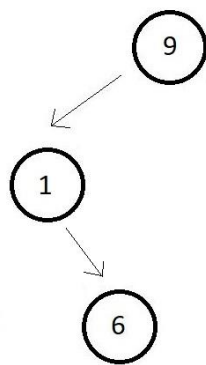
1

2

5. **Skewed trees:**

Skewed trees are binary trees where every parent node has just a single child and that child should be strict to the left or to the right for all the parents. Slewed trees having all the child nodes aligned to the left are called left-skewed trees, and skewed trees having all the child nodes aligned to the right are called right-skewed trees. Examples of both left and right-skewed trees are given below.
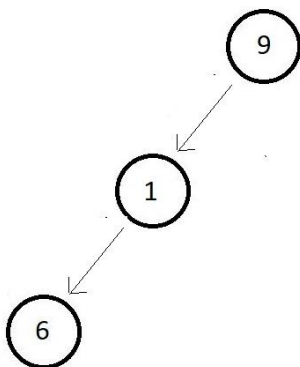
Left Skewed Tree
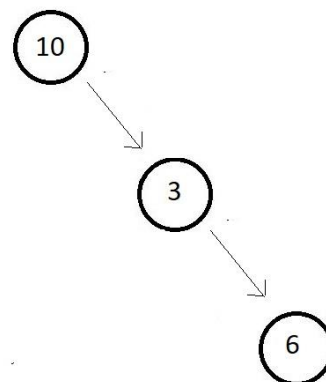
Right Skewed Tree

And those were all the various types of binary trees we had. You really don't have to memorize them all, rather just prepare some notes for now, and keep them for your interviews/exams. We already have quite a lot to keep in mind, and I'll not further stress you out by making you memorize these. Keep your concepts straight and you can learn things as per your requirement.