Hierarchical Data Structures.

Ortiz Vega Angelo
Course Code: CE1103
Name: Algorithms and Data Structures I,
Academic Area of Computer Engineering.
Cartago, Costa Rica.

-Keywords: Hierarchical Data Structures, Binary Tree, AVL Tree, Graphs, Heap.

-Content:

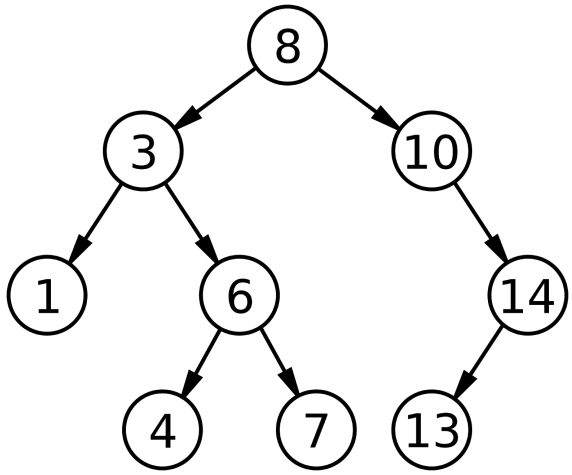
Unlike linear data structures, there are levels of hierarchy between the elements.

The trees seek to combine the advantages of arrays and linked lists: insertions and quick deletions, in addition to quick searches.

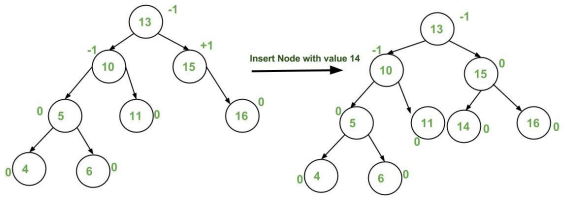
A tree consists of nodes and edges. They are usually small up and large as they approach the leaves. A leaf is a node without children, a tree has an underline and each node is on a level.

Types of Trees:
~ Binaries
~ Search Binaries
~ AVL
~ Splay
~ N-arios

1. **Binary Search Tree:**
A binary tree has at most at each node two children: left son and right son. It is called a binary search tree (BST) if the left child is younger than the father and the right child is older than the father.



2. AVL Tree:
AVL: Adelson-Velskii and Landis. It is a binary search tree with a special condition to ensure that the depth is optimal.
1. The height on the left can not differ in more than one unit with respect to the one on the right.
2. Height is the level plus one.

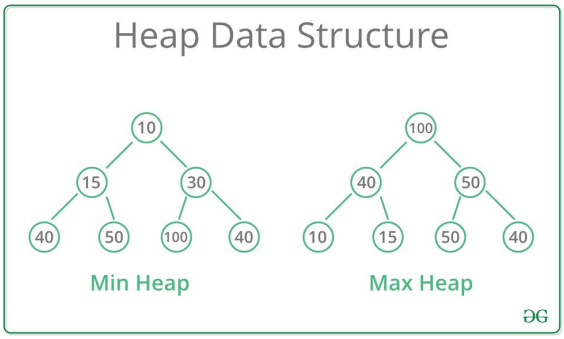


3. Heap:

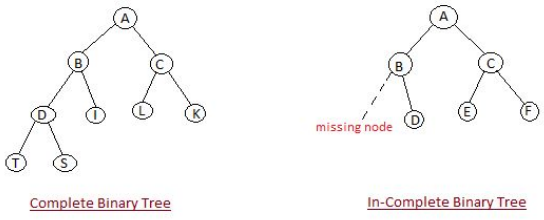
It is a special type of tree that allows you to implement priority queues, it is not the same as the memory heap. A heap is a binary tree with the following characteristics:

1. It is complete.
2. It is implemented as an array.
3. It satisfies the condition of Heap.

That it is complete means that each level is full when reading it from left to right except for the last level.



Complete vs Incomplete Tree.



4. Graphs:

They are general purpose data structures with a wide range of practical applications. A graph G is composed of:

1. Vertices: Nodes that represent entities.
2. Edges: Arcs or lines that represent relationships between entities.

G = {V, A}