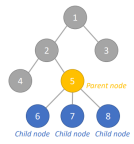
**Trees:**

A tree is a structure composed of **nodes** (that have keys, and eventually payloads) connected by **edges**.

The first node of a tree is called the root node. 

A node is a parent if it has connections with more than 1 edge,

and it’s always a child except for the root node.

Children nodes of the same parent are called sibling nodes.

A node with no child is called a leaf node.

**Properties:**

| **Name** | **What does it indicate** |
| --- | --- |
| **Level** | It denotes the number of edges on the path from the root to an edge |
| **Height** | Is equal to the maximum level of any node in the tree |
| **Height**  **balances** | A binary tree is height-balanced if the height of the left and right subtree of any node differ by not more than 1 |
| **Perfectly**  **height-balance d** | A binary tree is perfectly height-balanced if the height of the left and right subtree of any node is the same |

**Complete** A binary tree is complete if all its levels are completely filled, except the last one (which must be filled from left to right)

A tree can be **parsed**, that is when the order of the nodes has a meaning, for example by reproducing a mathematical expression.

A tree can also be **min/max heap**, that for the min one is when it is complete and the key of the parent node is always smaller than the one of the child node.

**Traversing a tree:**

There are **3 ways** to traverse a tree:

● Pre-order

● In-order

● Post-order