DEFINITIONS

- A graph is said to be **circuit-free** iff it has no non-trivial circuits.
- A graph is called a **forest** iff it is circuit free. A forest is a collection of trees
- A graph is called a **tree** iff it is circuit-free and connected.
- A graph that does not have any vertices or edges is called a **zero-order graph** and is not a tree. However it is a forest with 0 trees.
- A **trivial tree** is a graph that consists of a single vertex.
- A vertex of degree 1 in a tree is called a **terminal** vertex or a **leaf**. A vertex of degree >1 in a tree is called an **internal vertex** or **branch vertex**.

RECURSIVE DEFINITIONS

For any natural number *n*:

A graph built by connecting a single vertex *v* to *n* separate trees $T_1, \ldots, T_n$ by adding an edge between *v* and one of the vertices of each of $T_1, \ldots, T_n$ is a tree.

PROPERTIES

- For any positive integer *n*, any connected graph with *n* vertices is a tree iff it has $n-1$ edges.
- Corollary: Any tree that has more than one vertex has at least one vertex of degree 1.

ROOTED TREES

- A **rooted tree** is a tree in which one vertex is distinguished from the others and is called the **root**.
- The **level** of a vertex in a rooted tree is the number of edges along the unique path between it and the root. The root is at level 0.
- The **height** of a rooted tree is the maximum level of any vertex in that tree.
- Given any internal vertex *v* of a rooted tree (including the root), the **children** of *v* are all the vertices of the tree that are adjacent to v and one level farther away from the root than *v*. If *w* is a child of *v*, then *v* is called a **parent** of *w*. Two vertices that have the same parent are called **siblings**.

- For any vertex *v* of a rooted tree other than the root, the **ancestors** of *v* are all the vertices in the path between *v* and the root, including the root. If a vertex *v* is an ancestor of a vertex *w*, then *w* is a **descendant** of *v*.

BINARY TREES

- A **binary tree** is a rooted tree in which every parent has at most two children. Each child in the tree is designated as either the **left child** or **right child**, and every parent has at most one left child and one right child.
- A **full** binary tree is a binary tree where each parent has exactly two children.
- A **complete** binary tree is a binary tree where all levels are full except possibly for the last.
- A **perfect** binary tree is a binary tree where all levels are full
- Give any parent v in a binary tree T, if v has a left child, then the **left subtree of v** is the binary tree whose root is the left child of v, whose vertices consist of the left child of v and all its descendants, and whose edges consist of all those edges of T that connect the vertices of the left subtree. The **right subtree of v** is defined analogously.
- If k is a positive integer and T is a full binary tree with k internal vertices, then T has a total of 2k+1 vertices, including k+1 terminal vertices.
- For all integers $h \geq 0$ if Tis any binary tree of height h and with t terminal vertices, then $t \leq 2^h$, i.e. $\log_2 t \leq h$

SPANNING TREES AND WEIGHTED GRAPHS

- A **spanning tree** for a graph G is a subgraph of G that contains every vertex of G and is a tree.
- Every connected graph has a spanning tree
- A **weighted graph** G is a graph for which each edge e has an associated positive real weight w(e). The sum of all the weights of all the edges is the total weight of the graph, w(G).
- A **minimum spanning tree** for a connected weighted graph is a spanning tree that has the least possible total weight compared to all possible spanning trees for the graph.