Graphs: A collection of nodes and edges. We use \( V \) to denote the nodes and \( E \) the edges.

\[
G = (V, E)
\]

Directed graphs:

\[
\begin{align*}
1 &\rightarrow 2 \\
1 &\leftarrow 2
\end{align*}
\]

Examples:

- Transportation Network. (e.g. nodes = airports, edges = non-stop routes)
- Communication Network
- Social Network

Paths: A path \( P = (v_1, v_2, \ldots, v_k) \) in \( G = (V, E) \) is a sequence of nodes where each consecutive pair \( (v_i, v_{i+1}) \in E \).

Simple paths: A path \( P = (v_1, v_2, \ldots, v_k) \) is simple if all \( v_i \)'s are distinct.

Connected graphs: A graph \( G = (V, E) \) is connected iff there exist a path from \( u \) to \( v \).

Trees: A graph \( G = (V, E) \) is called a tree iff \( G \) is connected and \( G \) has no cycles.

Cycles: A cycle is a path \( (v_1, v_2, \ldots, v_k) \) where \( v_1 = v_k \) and the first \( k-1 \) nodes are distinct.

Claim 3.1: Every \( n \)-node tree has exactly \((n-1)\) edges.

Proof Sketch: Given a tree \( T \), root it at a node \( r \). For every node \( v \neq r \) there is a unique edge that is incident to \( v \) and point to \( r \).

Parent, Child, Descendant, Ancestor.
Q: Is G connected?

A: Breadth-First Search (BFS):
Depth-First Search (DFS):

BFS: Start at s ∈ V.
- Include all neighbors of s as the first layer.
- Include all unvisited neighbors of all first-layer nodes as the second layer. Repeat this process.

S = \{1\}

BFS tree: This BFS tree contains all nodes reachable from 1.

Suppose (u, v) ∈ E.
In the BFS tree, u is in layer i, v is in layer j.
|i - j| ≤ 1. (Claim (3.4) on page 81 of textbook)