Trees

CS 330: Discrete Structures
i.e. path from a vertex back to itself (w/ no repeating edges)

a **tree** is a connected, undirected graph w/ no simple circuits

a **forest** is a graph containing multiple, disconnected trees (aka. "acyclic graph")
e.g., tree, forest, or neither?
conjecture: if a graph is a tree, then there is a unique simple path between every pair of distinct vertices in the graph.

Assume this isn't true, then we can have a tree where:

- Paths split somewhere
- This creates a simple circuit
- Contradicts the definition of a tree
- Conjecture is true

(can you prove the converse of the conjecture?)
A rooted tree is a tree where one node is designated the "root", and all nodes can be characterized by its position relative to the root.

- Given two adjacent nodes in the tree, the node closer to the root is called the parent of the other, and the other is the child of the first.
- Sibling nodes share the same parent.
- There is a unique path from every node to the root.
  - The ancestors of a node $v$ are those nodes on the path from $v$ to the root.
  - Node $u$ is a descendant of $v$ if $v$ is on the path from $u$ to the root.
- nodes w/ children are called internal nodes
  - those n/ot are called leaf nodes (leaves)

- the depth of a node is the # of edges on the path from that node to the root

- the height of a node is the # of edges on the longest path from that node to a leaf

- the height of a tree is the height of the root
depth 0

height 5

depth 1

height 4

depth 3

height 2

depth 5

height 0

- internal nodes
- leaves

: migrainous:

• µ.

• matte:
an \textit{n-ary tree} is a tree where each node has no more than \textit{n} children

- an \textit{n-ary tree} up to \textit{n}=2 is a \textit{binary tree}

- a \textit{full n-ary tree} is one where every internal node has exactly \textit{n} children
e.g., draw full $n$-ary trees for $n = 2, 3, 4, 5$
Conjecture: a tree with $n$ nodes has $n-1$ edges.

PM I: Basic: $P(1)$ — Tree with 1 node has 0 edges $\checkmark$

Inductive step: assume $P(k)$ true (i.e.):

- Suppose we have a tree with $k+1$ nodes.
- If we remove a leaf node along with the edge connecting it to the graph, we have a tree with $k$ nodes, which by i.h., has $k-1$ edges.
- Adding it back increases the edge count by one, so tree with $k+1$ nodes has $k$ edges.

Q.E.D.
every connected graph contains a subgraph that is a tree which has the same nodes as the graph.

we call this a spanning tree

e.g. find spanning tree(s) in this graph:
In a weighted connected graph, the minimum spanning tree (MST) is the spanning tree that minimizes the sum of its edge weights.

e.g., what is the MST of this weighted graph?

(need at least 4 edges)
— lots of applications!
  e.g., efficient layout of comm/transport networks
Prim's MST algorithm

general idea: incrementally grow a tree by adding edges of min weight between a node in the tree and an unreachable one from the rest of the graph.

(if graph contains no duplicate weights, this will give us a unique MST)
 Kruskal's MST algorithm

general idea: grow a tree by adding edges of min weight that do not form a simple circuit up tree