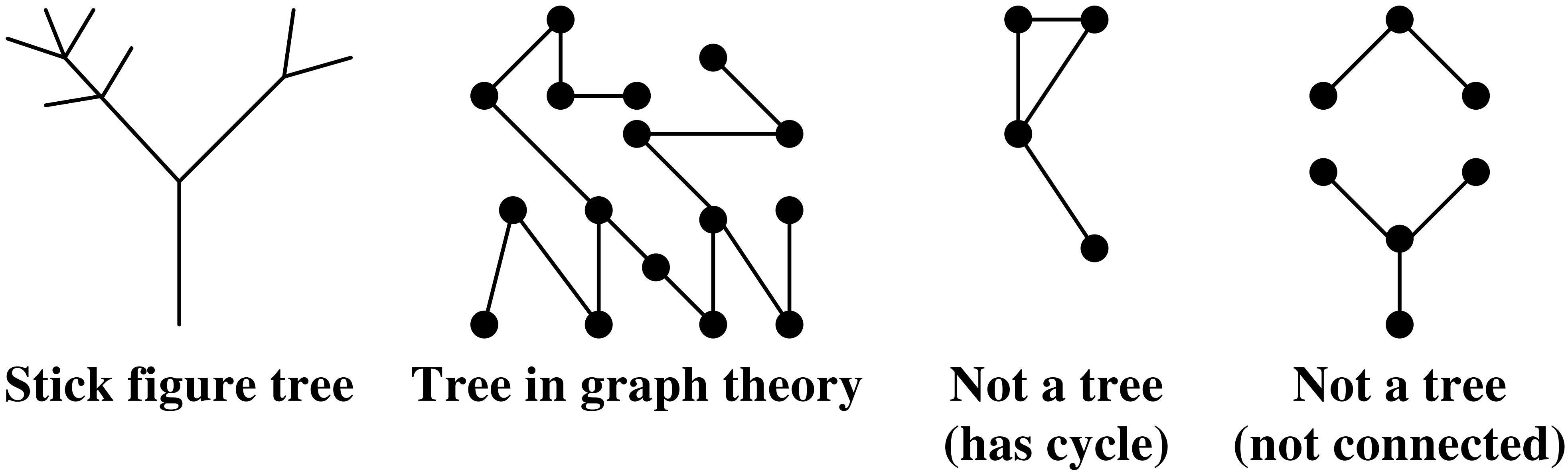
# Chapter 10.1
# Trees

Prof. Tesler

Math 184A
Winter 2017

# Trees



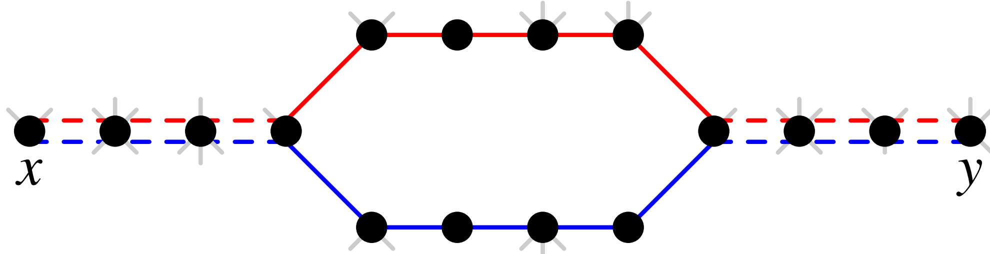

- A *tree* is an undirected connected graph with no cycles.
- It keeps branching out like an actual tree, but it is not required to draw it branching out from bottom to top.
- Genealogical trees, evolutionary trees, decision trees, various data structures in Computer Science

# Theorem:

## A tree has exactly one path between any pair of vertices

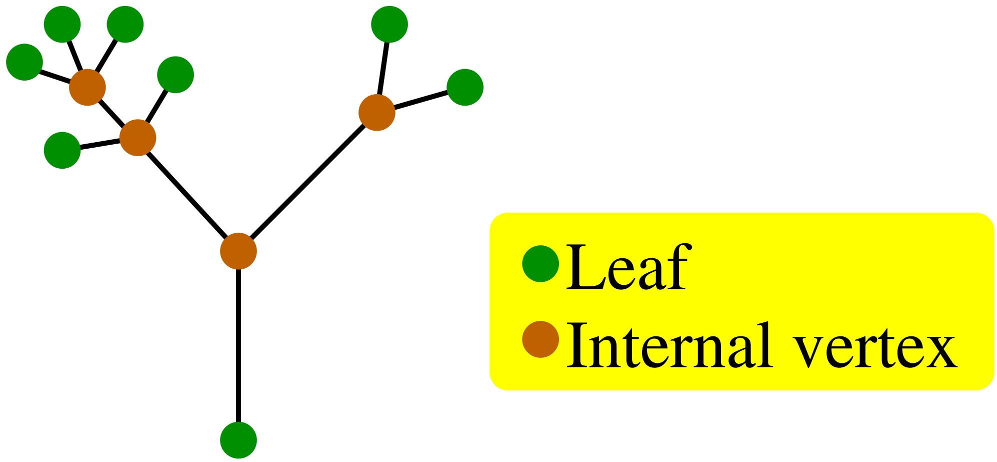
**Proof:**

- Let $x, y$ be any two distinct vertices.
- There is a path between them since the graph is connected.
- Suppose there are two unequal paths between them (red/blue).




- Superimposing the paths and removing their common edges (dashed) results in one or more cycles (solid).
- But a tree has no cycles!
  Thus, there cannot be two paths between $x$ and $y$.

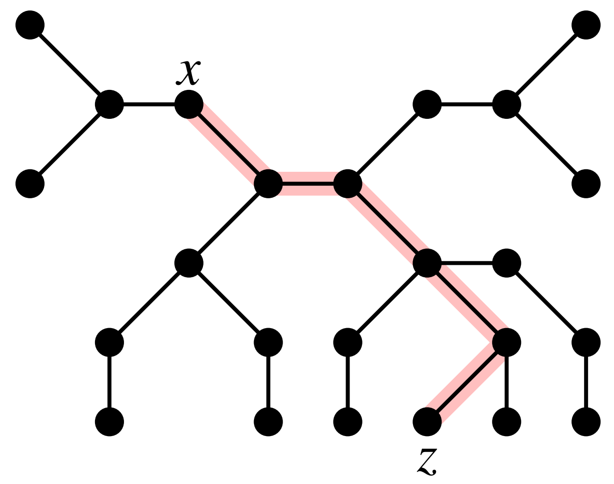
# Leaves




- A vertex of degree 1 is called a *leaf*.
  This tree has 8 leaves (including the bottom vertex).
- Sometimes, vertices of degree 0 are also counted as leaves.
- A vertex with degree $\geqslant 2$ is an *internal vertex*.
  This tree has 4 internal vertices.

# Theorem:

Every tree with at least two vertices has at least two leaves.

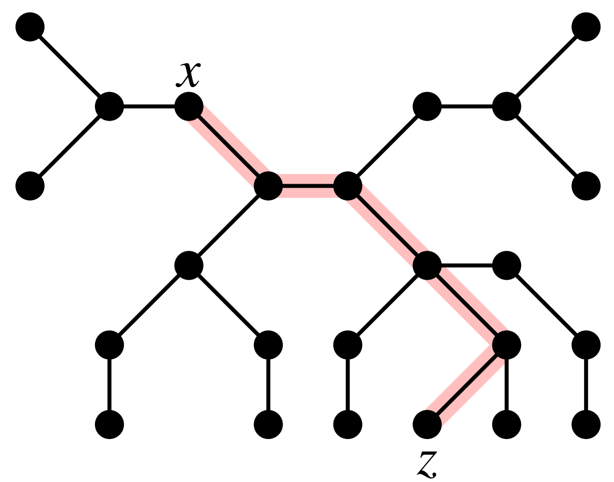



**Proof:**

- Pick any vertex, $x$.
- Generate a path starting at $x$:
  - Since there are at least two vertices and the graph is connected, $x$ has at least one edge. Follow any edge on $x$ to a new vertex, $v_2$.
  - If $v_2$ has any edge not yet on this path, pick one and follow it to a new vertex, $v_3$.
  - Continue until we are at a vertex $z$ with no unused edge.

# Theorem:

Every tree with at least two vertices has at least two leaves.

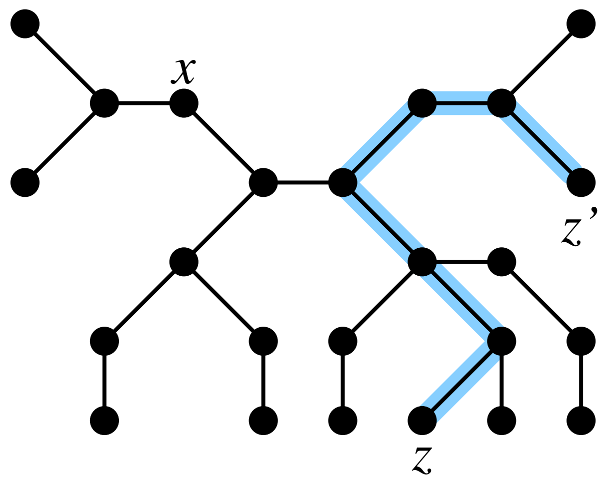



**Proof (continued):**

- There are no cycles in a tree, so $z$ cannot be a vertex already encountered on this walk.
- We entered $z$ on an edge, so $d(z) \geqslant 1$.
- We had to stop there, so $d(z) = 1$, and thus, $z$ is a leaf.

# Theorem:

Every tree with at least two vertices has at least two leaves.




**Proof (continued):**

- Now start over and form a path based at $z$ in the same manner; the vertex the path stops at is a second leaf, $z'$!

# Theorem:

All trees on $n \geqslant 1$ vertices have exactly $n-1$ edges

**Proof by induction:**

*Base case:* $n=1$

- The only such tree is an isolated vertex.
- This is $n=1$ vertex and no edges. Indeed, $n-1=0$.

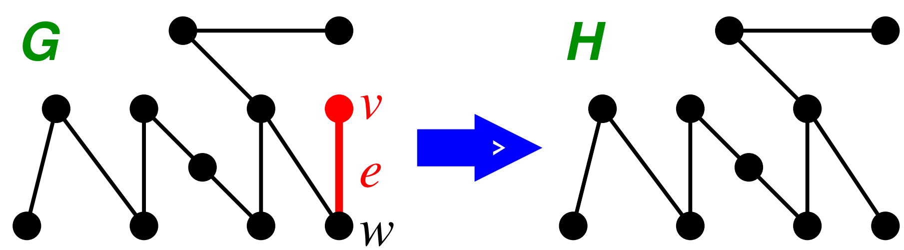
# Theorem:

All trees on $n \geqslant 1$ vertices have exactly $n-1$ edges

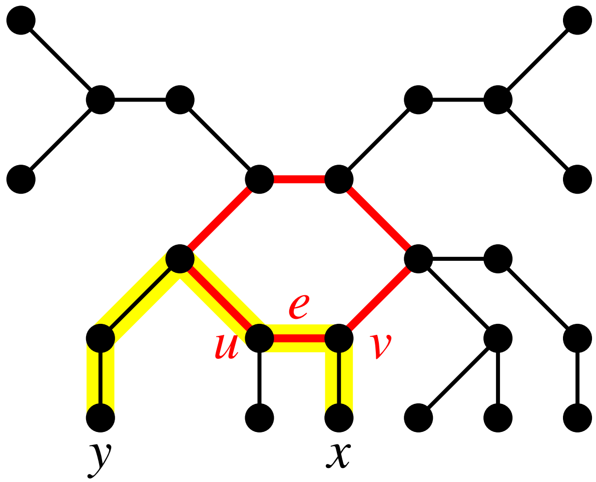
**Proof by induction (continued):**

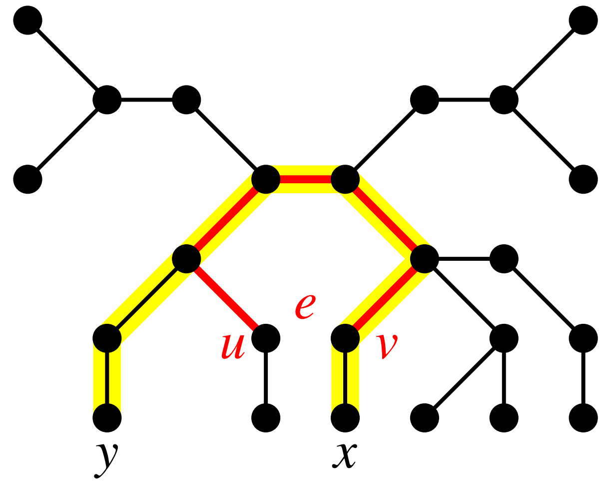
*Induction step:* $n \geqslant 2$. Assume the theorem holds for $n-1$ vertices.

- Let $G$ be a tree on $n$ vertices.
- Pick any leaf, $v$.
- Let $e = \{v, w\}$ be its unique edge.




- Remove $v$ and $e$ to form graph $H$:
  - $H$ is connected (the only paths in $G$ with $e$ went to/from $v$).
  - $H$ has no cycles (they would be cycles in $G$, which has none).
  - So $H$ is a tree with $n-1$ vertices.
  - By the induction hypothesis, $H$ has $n-2$ edges.
- Then $G$ has $(n-2)+1 = n-1$ edges.



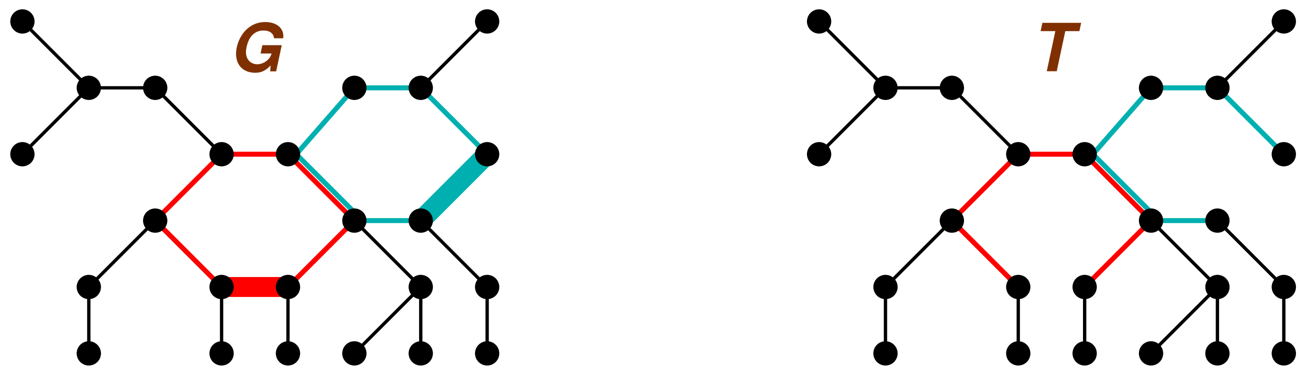




**Removing an edge from a cycle does not affect which vertices are in a connected component:**

- Consider a cycle (red) and edge ($e = \{u, v\}$) in the cycle.
- **Left graph:** Suppose a path (yellow) from $x$ to $y$ goes through $e$.
- **Right graph:**
  - Delete $e$. This disrupts the yellow path.
  - But the cycle provides an alternate route between $u$ and $v$!
  - Reroute the path to substitute $e$ (and possibly adjoining edges) by going around the cycle the other way.

# Spanning trees

- A *spanning tree* of an undirected graph is a subgraph that's a tree and includes all vertices.
- A graph $G$ has a spanning tree iff it is connected:
  - If $G$ has a spanning tree, it's connected: any two vertices have a path between them in the spanning tree and hence in $G$.
  - If $G$ is connected, we will construct a spanning tree, below.
- Let $G$ be a connected graph on $n$ vertices.
- If there are any cycles, pick one and remove any edge. Repeat until we arrive at a subgraph $T$ with no cycles.




- $T$ is still connected, and has no cycles, so it's a tree! It reaches all vertices, so it's a spanning tree.

# Converse theorem:

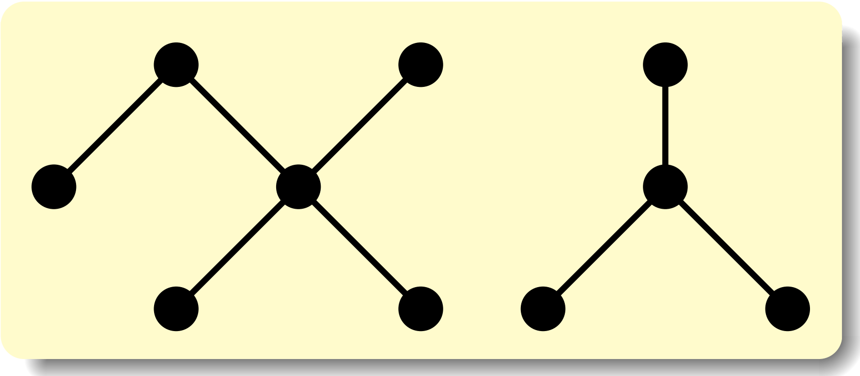
## If a connected graph on $n$ vertices has $n-1$ edges, it's a tree

**Proof:**

- Let $G$ be a connected graph on $n$ vertices and $n-1$ edges.
- $G$ contains a spanning tree, $T$.
- $G$ and $T$ have the same vertices.
- $T$ has $n-1$ edges, which is a subset of the $n-1$ edges of $G$. So $G$ and $T$ have the same edges.
- $G$ and $T$ have the same vertices and edges, so $G = T$. Thus, $G$ is a tree.

# Forest

- A *forest* is an undirected graph with no cycles.
- Each connected component is a tree.



| | # vertices | # edges |
|---|---|---|
| Left tree | 6 | 5 |
| Right tree | 4 | 3 |
| Total | 10 | 8 |

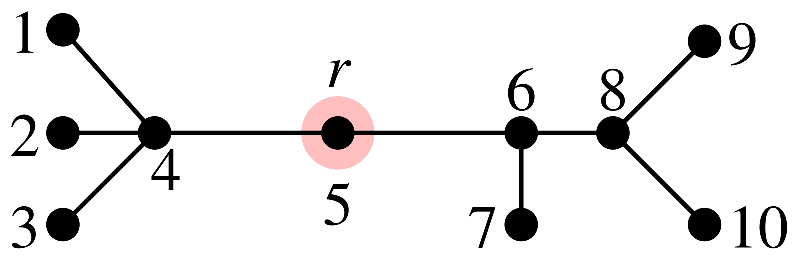
## Theorem

*A forest with $n$ vertices and $k$ trees has $n-k$ edges.*

## Proof

- The $i^{\text{th}}$ tree has $n_i$ vertices and $n_i - 1$ edges, for $i = 1, \ldots, k$.
- Let $n$ be the total number of vertices, $n = \sum_{i=1}^{k} n_i$.
- The total number of edges is $\sum_{i=1}^{k}(n_i - 1) = \left(\sum_{i=1}^{k} n_i\right) - k = n - k$

# Rooted trees




- Choose a vertex $r$ and call it the *root*. Here, $r = 5$ (pink).
- Follow all edges in the direction *away* from the root.
  - For edge $u \to v$, vertex $u$ is the *parent* of $v$ and $v$ is the *child* of $u$.
  - Children with the same parent are *siblings*.
- 5 is the parent of 4 and 6.
  4 and 6 are children of 5, and are siblings of each other.
- 4 is the parent of 1, 2, and 3.
  1, 2, and 3 are children of 4, and are siblings.

# Rooted tree examples

Rooted trees are usually drawn in a specific direction, e.g., bottom to top, top to bottom, left to right, or center to outside.

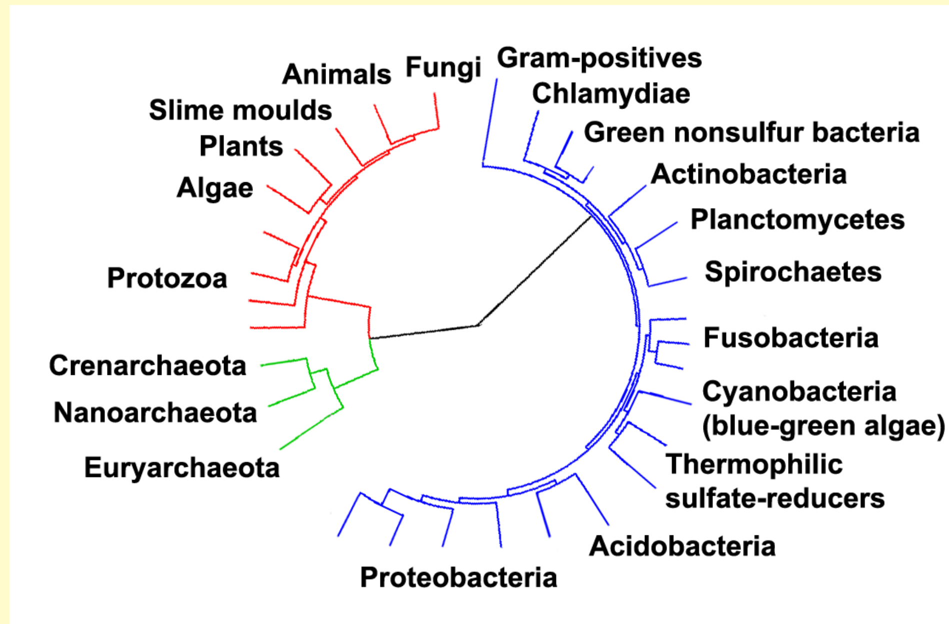
## Evolutionary trees

### Primates


http://en.wikipedia.org/wiki/File:PrimateTree2.jpg

**Root at bottom**
**Edges go bottom to top**

### Tree of Life




http://en.wikipedia.org/wiki/
File:Collapsed_tree_labels_simplified.png

**Root at center**
**Edges go out from center**