Trees

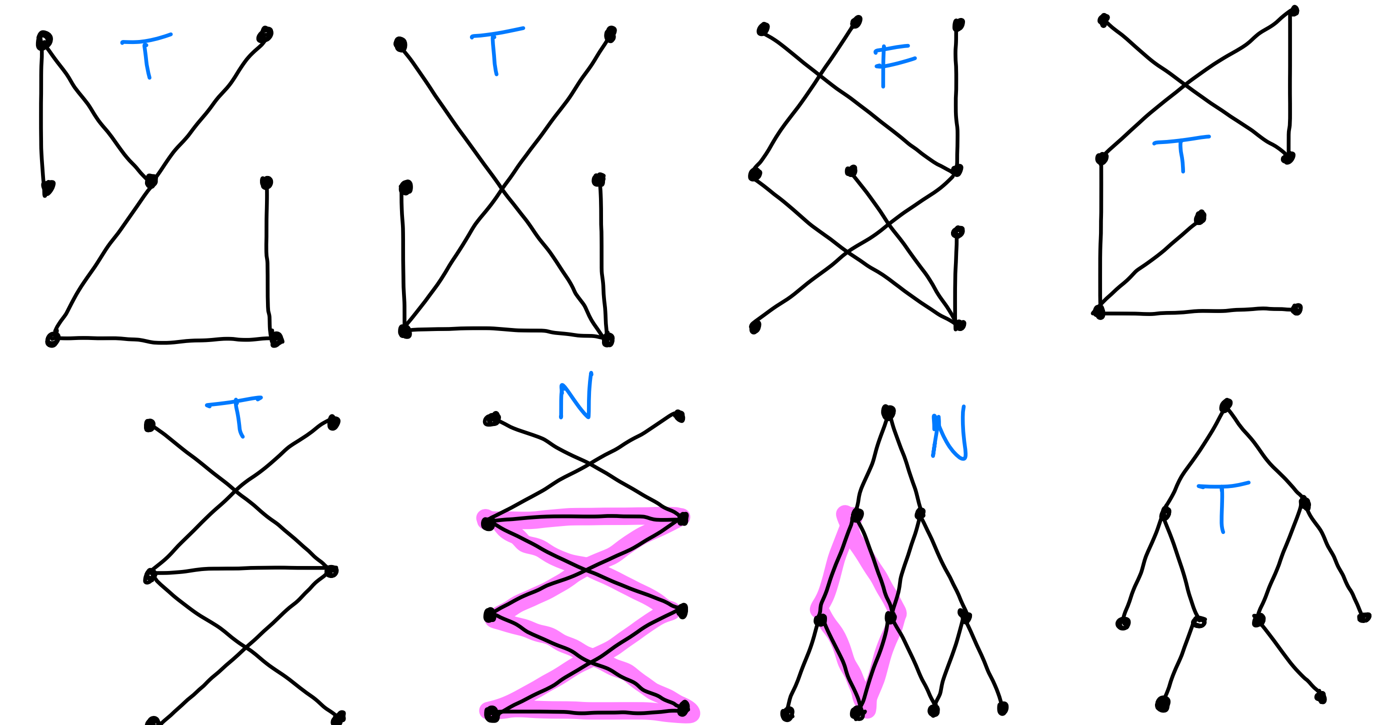
CS 330 : Discrete Structures

i.e. path from a vertex back to itself
(w/ no repeating edges)

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

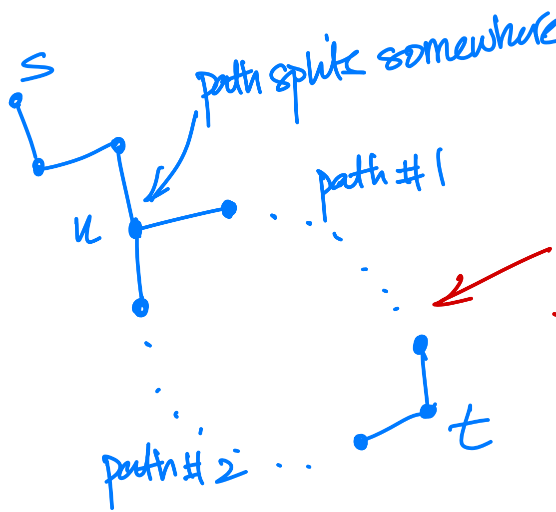
a **forest** is a graph containing multiple, disconnected trees
(aka. "acyclic graph")

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:



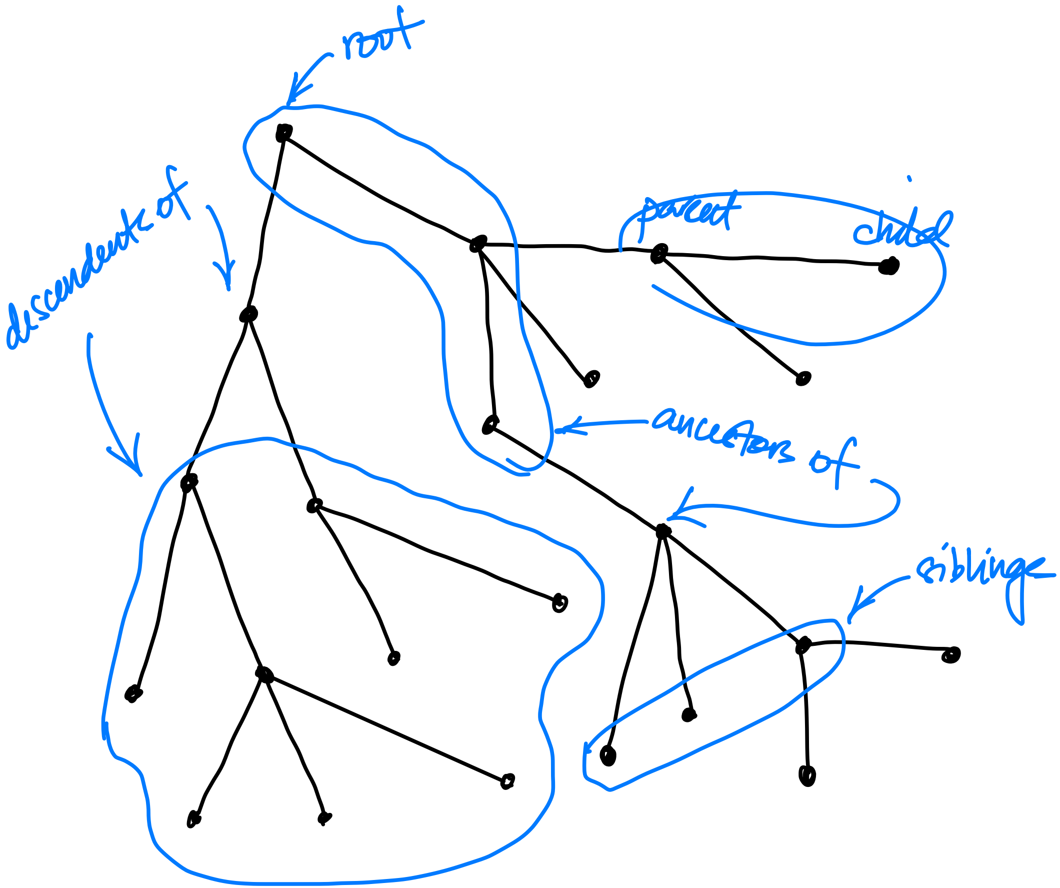
this creates a simple circuit!
— contradicts the definition of a tree

∴ conjecture is true

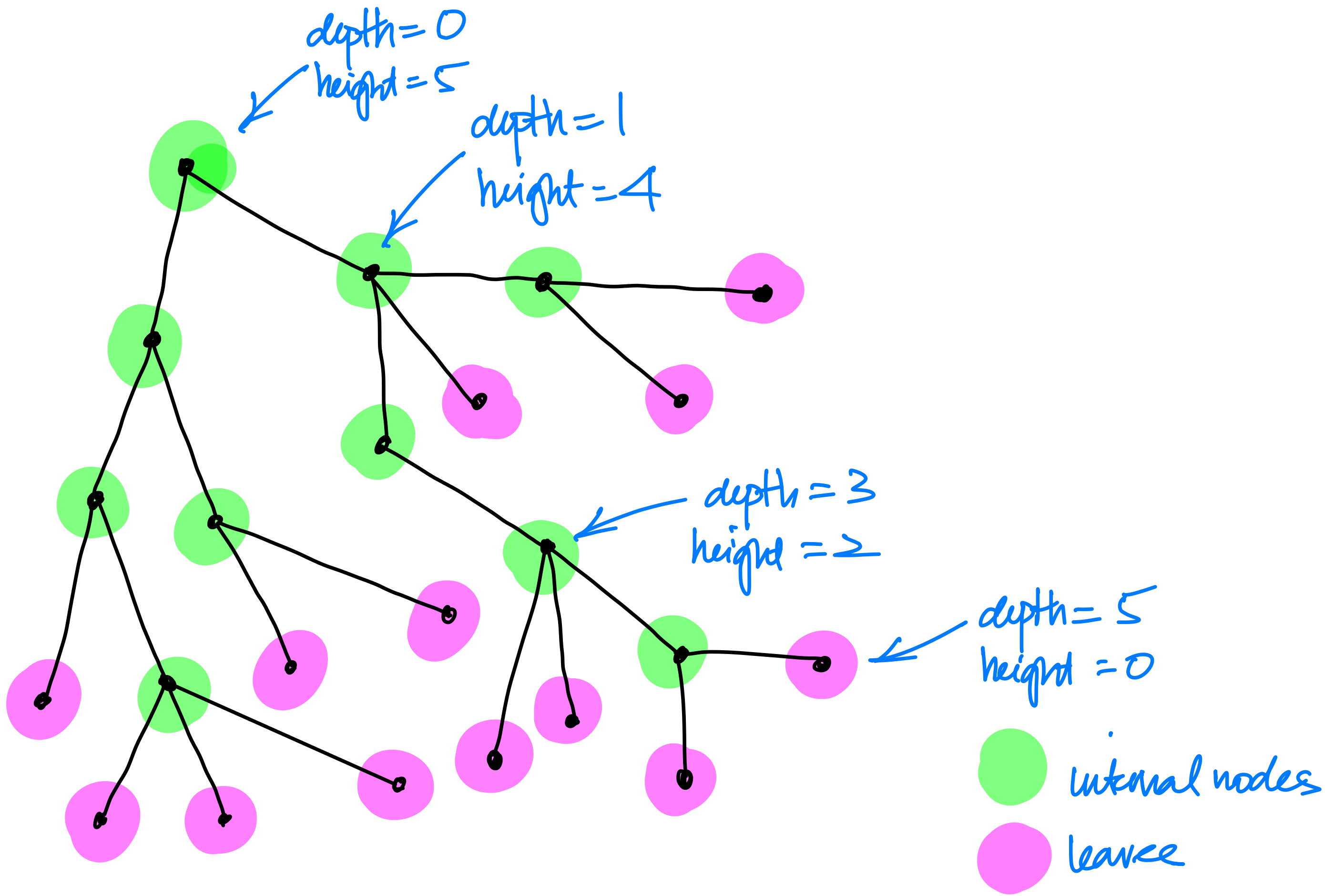
(can you prove the converse of the conjecture?)

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 — and cannot be adjacent!
- 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 descendent of v if v is on the path from u to the root


root
descendents of
parent
child
ancestors of
siblings

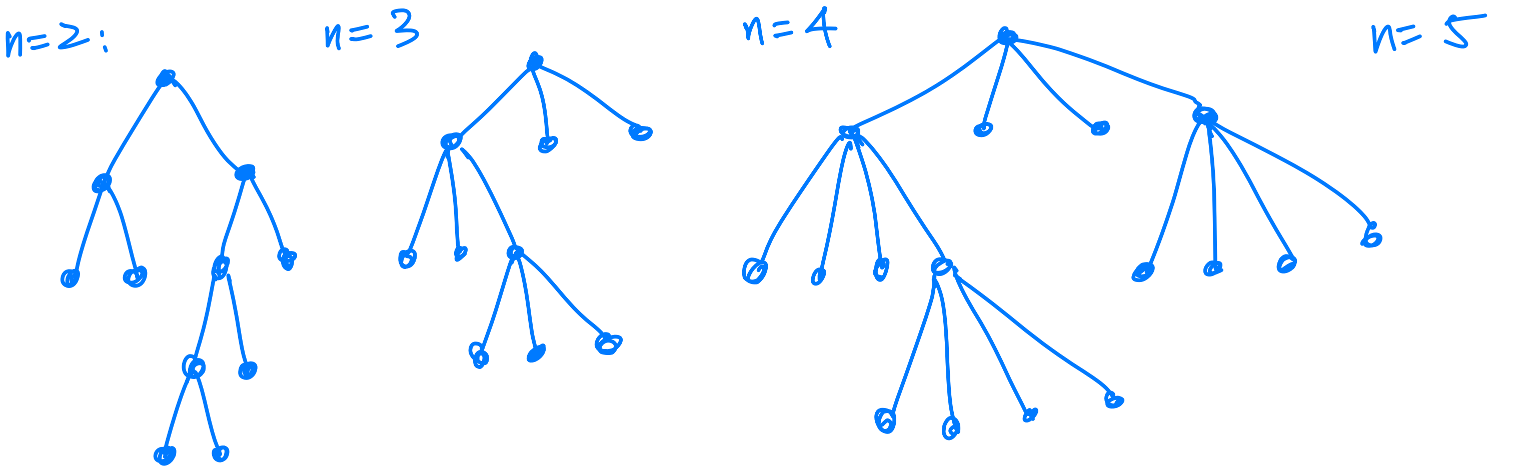
- nodes w/ children are called ==internal nodes==
  - those w/out are called ==leaf nodes== (leaves)
- the ==depth== of a node is the # of edges on the path from that node to the root
- the ==height== of a node is the # of edges on the longest path from that node to a leaf
  - the ==height of a tree== 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

an n-ary tree is a tree where each node has no more than n children

- an n-ary tree w/ n=2 is a binary tree
- a full n-ary tree is one where every internal node has exactly n children

e.g., draw full n-ary trees for n = 2, 3, 4, 5

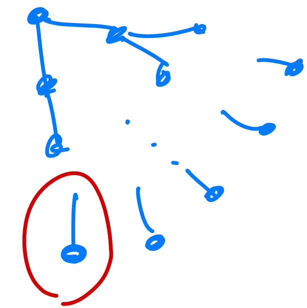


conjecture: a tree w/ n nodes has n-1 edges

PMI: basis: P(1) - tree w/ 1 node has 0 edges ✓

inductive step: assume P(k) true (I.H.)

- suppose we have a tree w/ k+1 nodes



- if we remove a leaf node along w/ the edge connecting it to the graph, we have a tree w/ k nodes, which by I.H., has k-1 edges.

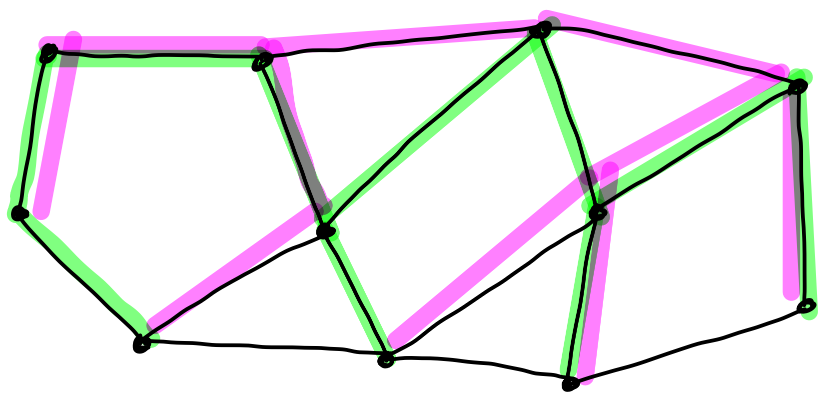
- adding it back increases the edge count by one, so tree w/ k+1 nodes has k edges

QED.

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

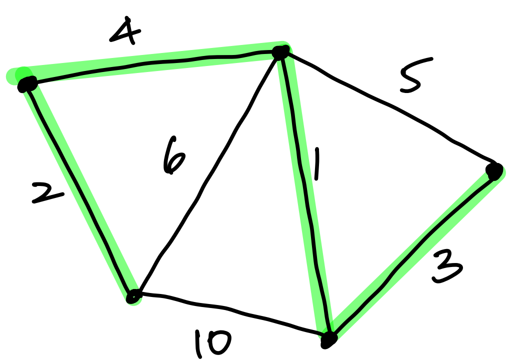
— we call this a spanning tree

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)

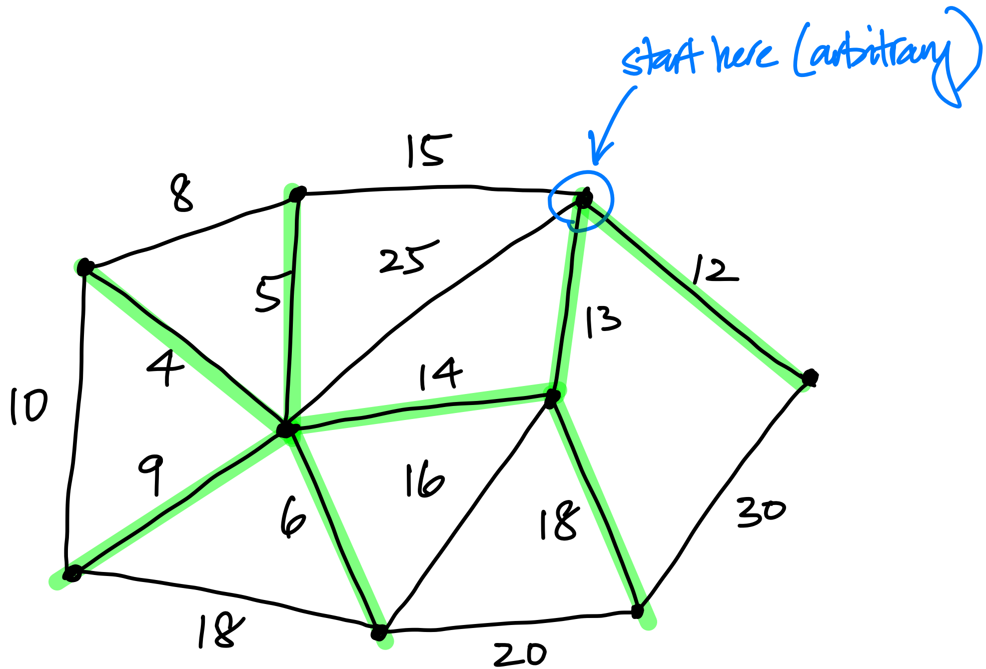
— lots of applications!

e.g., efficient layout of comm, transport networks

Prim's MST algorithm

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

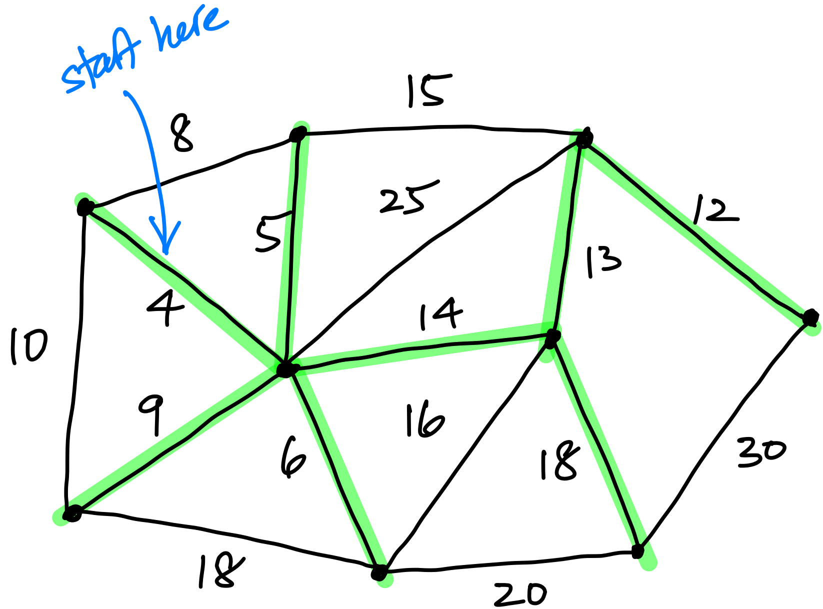
(if graph contains no duplicate weights, this will give us a <u>unique</u> MST)



start here (arbitrary)
15
8
25
5
12
13
4
14
10
9
16
6
18
30
18
20

Kruskal's MST algorithm

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


start here
8
15
25
5
4
12
13
14
10
9
16
6
18
30
18
20