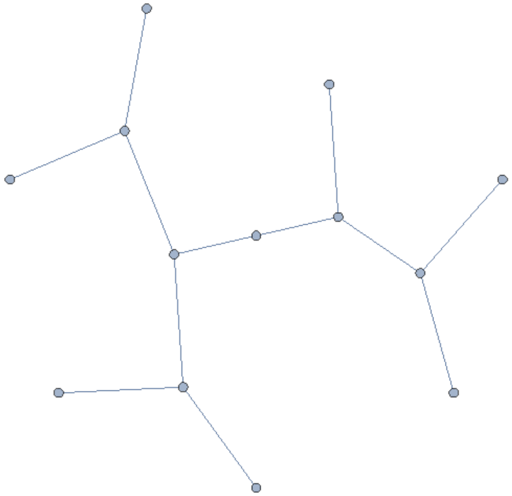
Minimum Spanning Trees

Warmups

A *tree* is defined to be a connected, undirected, acyclic graph.

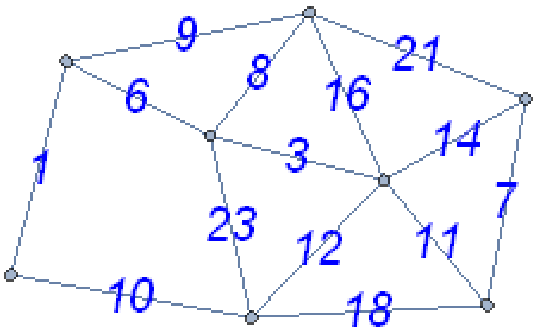
Claim: Any tree with at least one edge has at least one vertex with degree 1.
Proof: Assume that there is some tree T with at least one edge where all vertices have degree 2 or more. Take a walk in this tree. *What must eventually happen?*

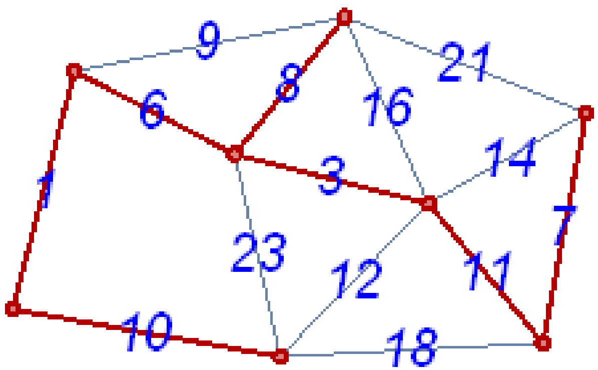
Claim: Any tree with n vertices contains exactly $n-1$ edges.
Proof: Math induction on n
How many edges does a tree with 1 vertex have? 0
Assume that all trees with k vertices contain exactly $k-1$ edges.
Consider any tree T with $k+1$ vertices. T must have at least one vertex v with degree 1. Let $T' \equiv T$ with v and its adjacent edge removed.
How many vertices does T' have? k
How many edges must T' have? $k-1$
How many edges must T have? k



Minimum Spanning Trees

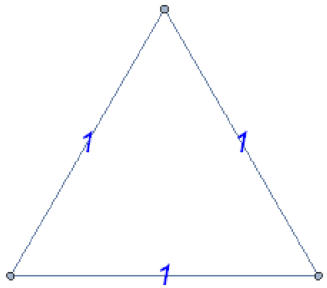
A *minimum spanning tree* of a connected, undirected, weighted graph is a sub-graph of minimum weight that is both a tree and spans the entire graph (i.e. touches every vertex).





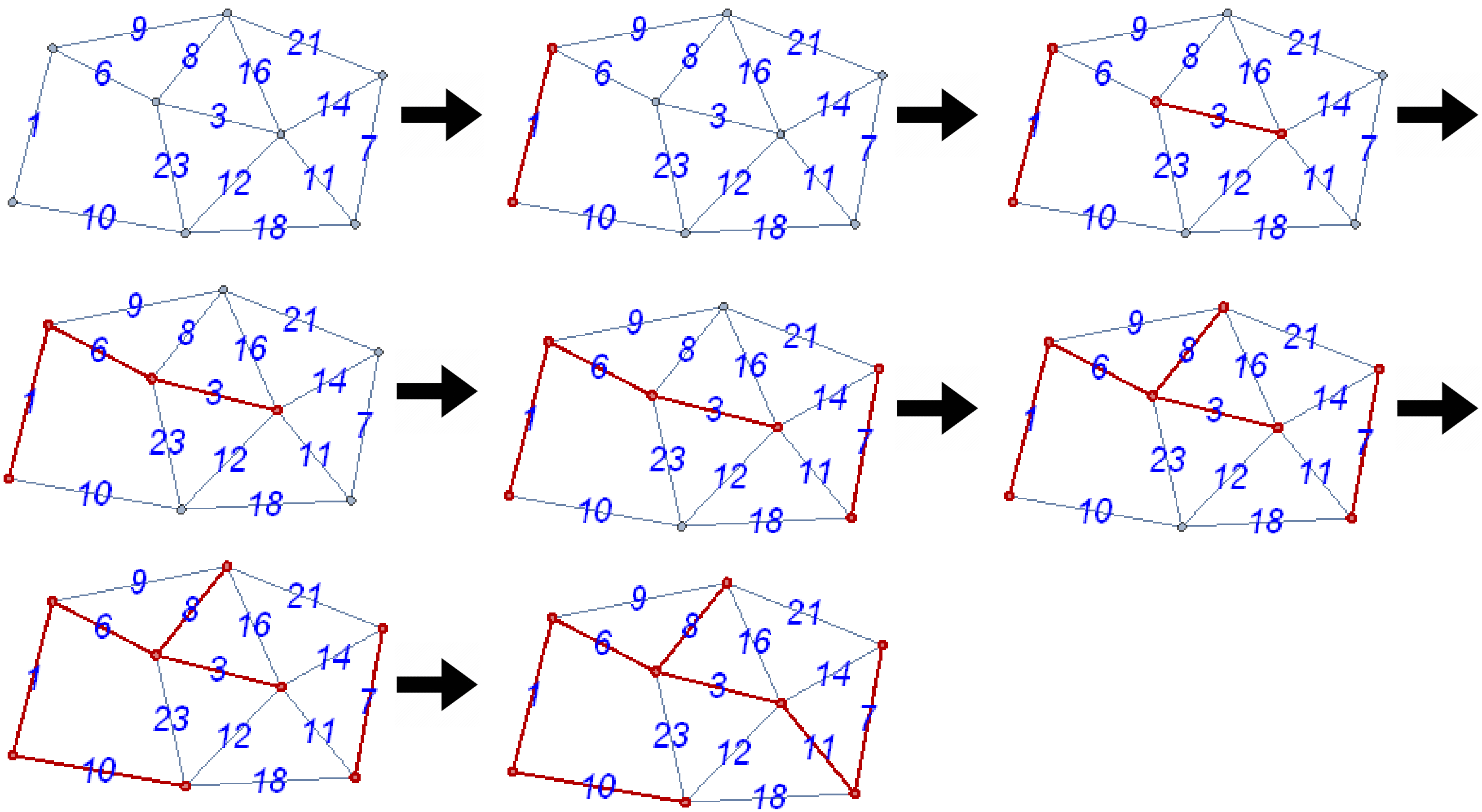
Applications: network design (roads, computers, etc.), broadcasting protocols, subroutine for more complex algorithms (CECS 428)

Note that the MST is *not necessarily* unique. For example, the following graph has 3 distinct solutions for the MST.



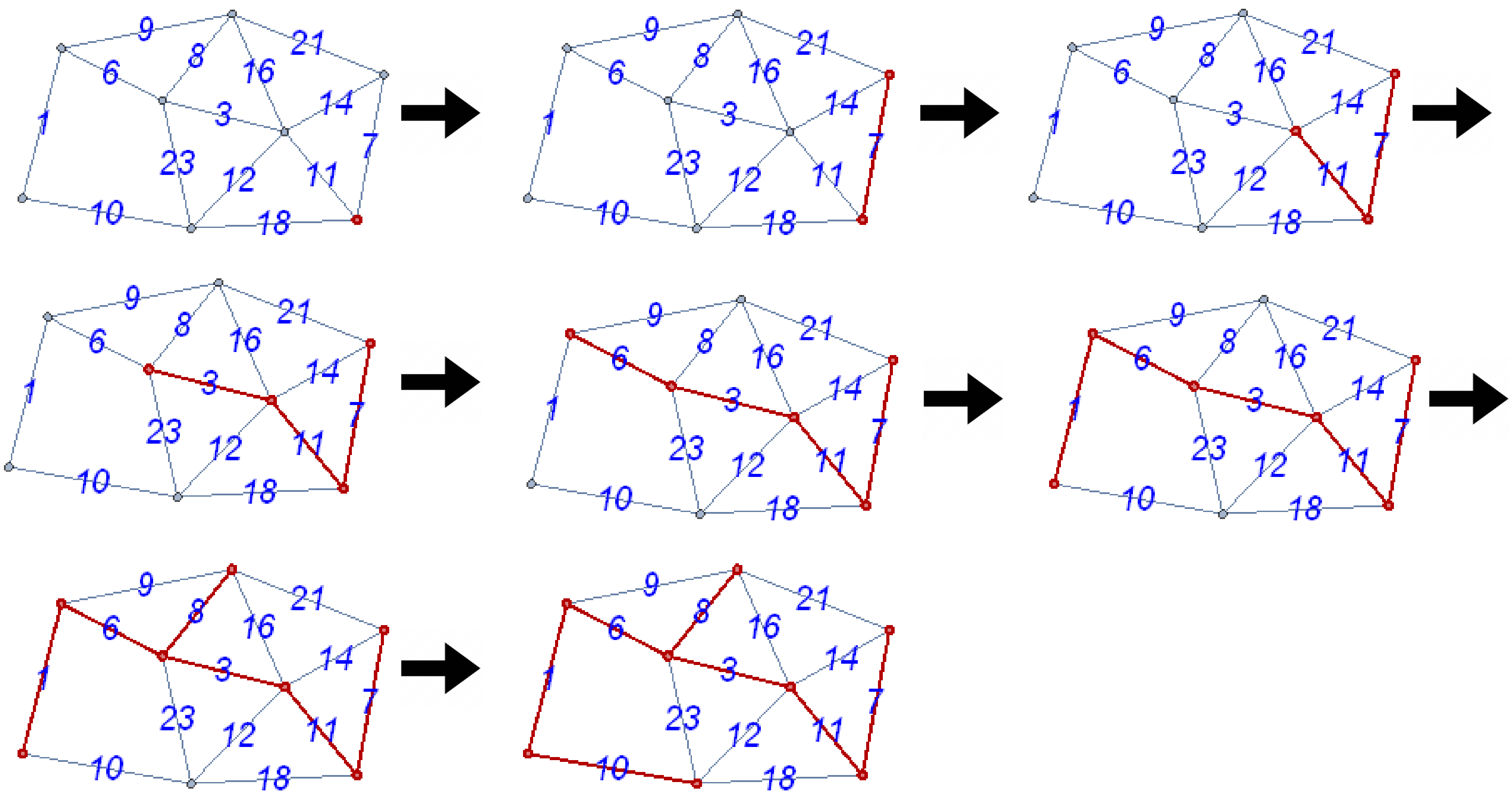
Kruskal’s Algorithm

Repeatedly choose to add the minimum weight edge left in the graph that does not create a cycle with the currently chosen edges until all edges have been considered.



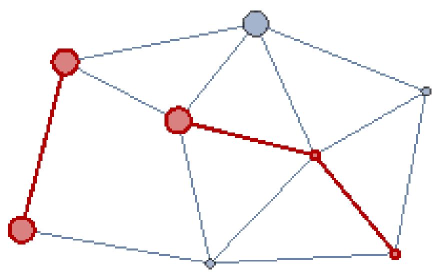
Prim’s Algorithm

Choose any starting vertex. Repeatedly add the minimum weight edge that is attached to the current tree and doesn’t create a cycle with the edges that have already been chosen until all edges have been considered.

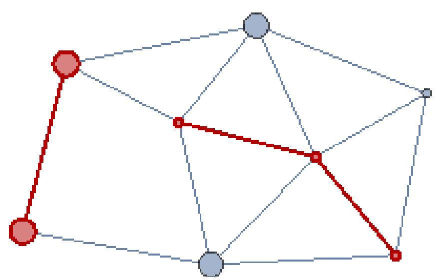


The Berkeley Lemma

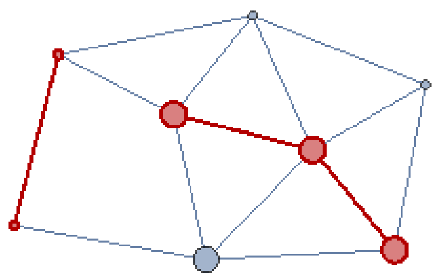
Lemma: Suppose edges X are part of a minimum spanning tree of $G = (V, E)$. Pick any subset of nodes S for which X does not cross between S and $V - S$, and let e be the lightest edge across this partition. Then $X \cup \{e\}$ is part of some MST.



Is this a legal selection for S? No



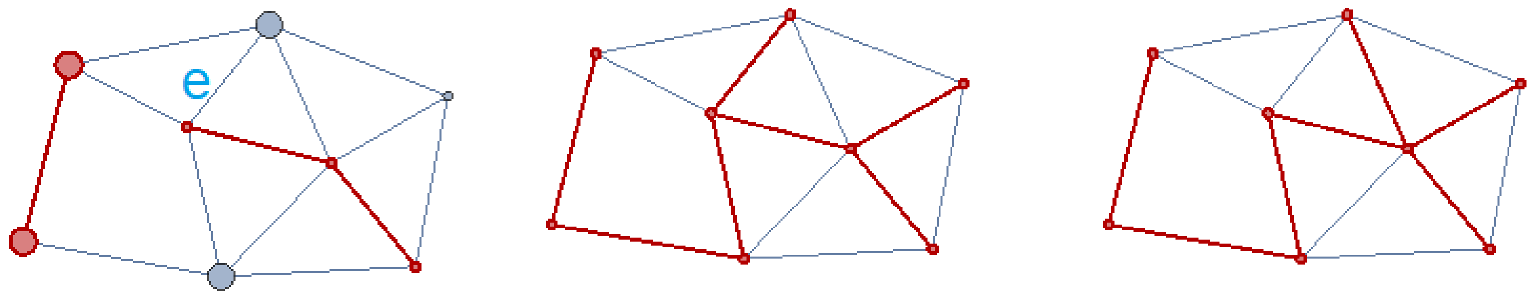
Is this a legal selection for S? Yes



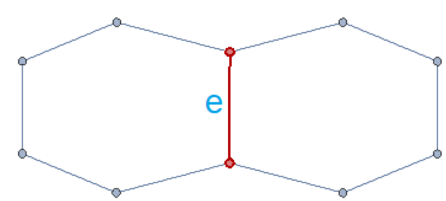
Is this a legal selection for S? Yes

The Berkeley Lemma

<u>Lemma</u>: Suppose edges X are part of a minimum spanning tree of $G = (V, E)$. Pick any subset of nodes S for which X does not cross between S and $V - S$, and let e be the lightest edge across this partition. Then $X \cup \{e\}$ is part of some MST.

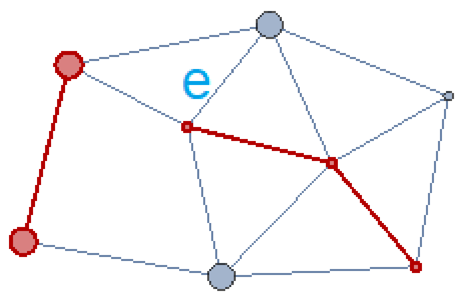


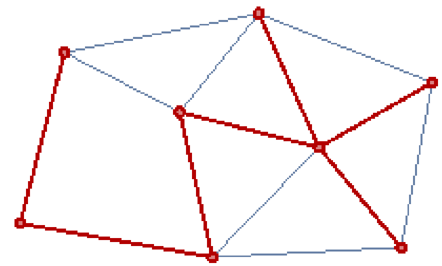
<u>Proof</u>: Let the edges X expand to MST T. If T contains the edge e, we're done.
Assume that T does not contain edge e.
Does e add any vertices to T? No.
Can a tree with a fixed number of vertices add an extra edge? No.
Is it possible for $T \cup \{e\}$ to be a tree? No.
Which of the requirements for being a tree does $T \cup \{e\}$ fail?
connected, undirected, acyclic
Can e have created more than one cycle in $T \cup \{e\}$? No.
$\rightarrow T \cup \{e\}$ has exactly one cycle.



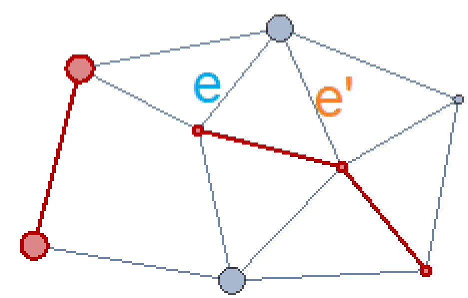
The Berkeley Lemma

<u>Lemma</u>: Suppose edges X are part of a minimum spanning tree of $G = (V, E)$. Pick any subset of nodes S for which X does not cross between S and $V - S$, and let e be the lightest edge across this partition. Then $X \cup \{e\}$ is part of some MST.





We know that $T \cup \{e\}$ has exactly one cycle.
If we start at the S vertex of the edge e and walk around the cycle to the other side of e, there must exist at least one edge that crosses from S to $V - S$ (because the other side of e is in $V - S$).
Let $e' \equiv$ be any edge that crosses from S to $V - S$ around that walk
Is $T - \{e'\} \cup \{e\}$ undirected? Yes.
Is $T - \{e'\} \cup \{e\}$ connected? Yes.
Is $T - \{e'\} \cup \{e\}$ acyclic? Yes.
$\rightarrow T - \{e'\} \cup \{e\}$ is a tree.
What can you say about $w(e)$ and $w(e')$? $w(e) \leq w(e')$
What can you say about $w(T)$ and $w(T - \{e'\} \cup \{e\})$?
$w(T - \{e'\} \cup \{e\}) \leq w(T) \Rightarrow w(T - \{e'\} \cup \{e\}) = w(T)$



Kruskal and Prim

Lemma: Suppose edges X are part of a minimum spanning tree of $G = (V, E)$. Pick any subset of nodes S for which X does not cross between S and $V - S$, and let e be the lightest edge across this partition. Then $X \cup \{e\}$ is part of some MST.

Kruskal's Algorithm: Repeatedly choose to add the minimum weight edge left in the graph that does not create a cycle with the currently chosen edges until all edges have been considered.
Choice: Consider the minimum weight edge $\{v_1, v_2\}$ left in the graph that does not create a cycle with the currently chosen edges. Choose to let $v_1 \in S$. Are we now *required* to put $v_2 \in \mathrm{S}$? No. HINT: If we are, then by the definition of S, $\{v_1, v_2\}$ creates a cycle with the currently chosen edges.

Prim's Algorithm: Choose any starting vertex. Repeatedly add the minimum weight edge that is attached to the current tree and doesn't create a cycle with the edges that have already been chosen until all edges have been considered.
Choice: Choose S to be all vertices in the current structure.

Implementation

- What data structure would be ideal to hold the weights of the edges remaining to be considered?

- According to the consensus on the internet, a simple CECS 328 implementation of Prim's might run in time $\sim |V|^2$ and a simple CECS 328 implementation of Kruskal's algorithm might run in time $\sim |E| \log |E|$. Thus, if the graph is sparse ($|E| \sim |V|$), Kruskal's algorithm is better, but if the graph is dense ($|E| \sim |V|^2$), Prim's is better.

- There is a randomized algorithm that runs in *linear* time in the number of edges. (This is spooky.)