Minimum Spanning Trees
CS 320, Fall 2017

Dr. Geri Georg, Instructor
georg@colostate.edu

Some materials adapted from Prof. Wim Bohm
Algorithmically....

Given a set of locations, with *positive* weights of some sort, we want to create a network \((T)\) that connects all nodes to each other with minimal sum of distances \((c_e, c \in T)\).

\[ G = (V, E) \]

\[ \sum_{e \in T} c_e = 50 \]
An Example
An Example
An Example
Clustering
Biological Prediction

neuron

dendrites

dendrite length predicted using MSTs

real dendrites

MST predictions
Greedy MST Algorithms

**Kruskal's algorithm.** Start with \( T = \emptyset \). Consider edges in ascending order of cost. Add edge \( e \) to \( T \) unless doing so would create a cycle.

**Reverse-Delete algorithm.** Start with \( T = E \). Consider edges in descending order of cost. Delete edge \( e \) from \( T \) unless doing so would disconnect \( T \).

**Prim's algorithm.** Start with some node \( s \) and greedily grow a tree \( T \) from \( s \). At each step, add the cheapest edge \( e \) to \( T \) that has exactly one endpoint in \( T \), i.e. without creating a cycle.
Cut Property

Let $S$ be a subset of nodes, and let $e$ be the min cost edge with exactly one endpoint in $S$. Then the MST $T$ contains $e$.

Proof: $T$ is a spanning tree so it has to have a path from $v$ to $w$. Follow that path to $w$. It has to go somehow from $v$ to $v'$ (last node on $P$ in $S$), to $w'$ (1st node on $P$ in $V-S$), then somehow to $w$. 
Cut Property

Let $S$ be a subset of nodes, and let $e$ be the min cost edge with exactly one endpoint in $S$. Then the MST $T$ contains $e$.

Proof continued: Now exchange $e$ for $e'$ to get:

$$T' = T - \{e'\} \cup \{e\}$$

We claim $T'$ is a spanning tree. It is connected since any path can use $e$ instead of $e'$. It is acyclic since we substituted $e$ for $e'$. $e$ is the cheapest edge with just 1 node in $S$, so $c_e < c_{e'}$, and the cost of $T'$ < the cost of $T$. 
Kruskal’s Algorithm

Create sets for each tree in the forest – initially one for each \( u \in V \)

Sort edges from least cost to most

For each edge \((u,v)\) taken in increasing cost order, if \( u \) and \( v \) are in different trees, add the edge to the MST and merge the vertices of the 2 trees
Muddy City – Kruskal’s
For each edge \((u,v)\) taken in increasing cost order, if \(u\) and \(v\) are in different trees, add the edge to the MST and merge the vertices of the 2 trees.

<table>
<thead>
<tr>
<th>ij</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>cd</td>
<td>2</td>
</tr>
<tr>
<td>be</td>
<td>2</td>
</tr>
<tr>
<td>hi</td>
<td>2</td>
</tr>
<tr>
<td>gk</td>
<td>2</td>
</tr>
<tr>
<td>jk</td>
<td>2</td>
</tr>
<tr>
<td>bf</td>
<td>3</td>
</tr>
<tr>
<td>bc</td>
<td>3</td>
</tr>
<tr>
<td>ef</td>
<td>3</td>
</tr>
<tr>
<td>af</td>
<td>3</td>
</tr>
<tr>
<td>hk</td>
<td>3</td>
</tr>
<tr>
<td>ei</td>
<td>3</td>
</tr>
<tr>
<td>de</td>
<td>3</td>
</tr>
<tr>
<td>bd</td>
<td>4</td>
</tr>
</tbody>
</table>

\(a\) \(b\) \(c\) \(d\) \(e\) \(f\) \(g\) \(h\) \(i\) \(j\) \(k\)

\(a-b, b-c, c-e, e-f, f-d, d-j, j-i, i-a\)

\(1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-9, 9-10\)
Prim’s Algorithm

Initialize $S = \text{any node}$. This will be the MST root.

Apply cut property to $S$: add min cost edge $(v, w)$ where $v$ is in $S$ and $w$ is in $V-S$, and add $w$ to $S$.

Repeat until $S = V$. 
Prim(G,r)

    
    foreach (u ∈ V)
            a[u] ← ∞; u.π = Ø
    a[r]= 0

min priority queue Q = {}

    foreach (u ∈ V) insert u onto Q (key: ‘a’
            value)

set S ← {}

while (Q is not empty) {
        u ← extract min element from Q
        S ← S ∪ { u }

    foreach (edge e = (u, v) incident to u)
        if ((v ∉ S) and (c_e < a[v]))
            decrease priority a[v] to c_e
            v.π = u
Muddy City – Prims’s

Graph showing connections between different points with edge weights.
min priority queue \( Q = \{ \} \)

\[
\text{foreach } (u \in V) \text{ insert } u \text{ in } Q
\]

set \( S \leftarrow \{ \} \)

while (\( Q \) is not empty) {
   \[
   u \leftarrow Q \text{ min element (delete)}
   \]
   \[
   S \leftarrow S \cup \{ u \}
   \]
   \[
   \text{foreach edge } e = (u, v)
   \]
   \[
   \text{if } ((v \not\in S) \text{ and } (c_e < a[v]))
   \]
   \[
   a[v] = c_e
   \]
   \[
   v.\pi = u
   \]

Causes \( Q \) to change
Image Credits


muddyCity: http://www.utdallas.edu/~besp/teaching/mst-applications.pdf

clusteringMST2: https://www.researchgate.net/figure/236106483_fig2_Figure-2-Unrooted-minimum-spanning-tree-network-showing-genetic-relationship-among

dendritePP: http://www.pnas.org/content/109/27/11014.full.pdf