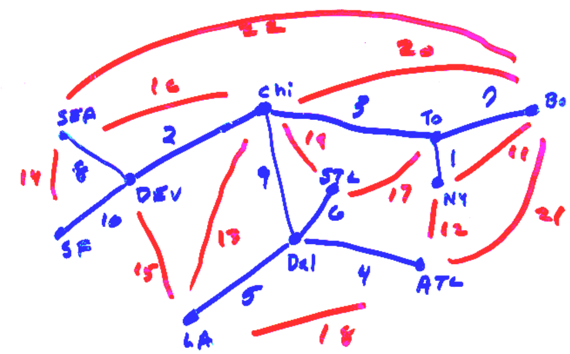
Minimum Spanning Trees

Kruskal's Algorithm:

An Example of the Greedy Method

Minimum Cost Spanning Tree



Given: Undirected graph $G = (V, E)$

Edge cost/weight $c(e) \in \mathbb{R}$

Find: subset $T \subseteq E$

s.t. $\sum_{e \in T} c(e) = c(T)$

is minimal

subject to: (V, T) connected

FACT:

Adding an edge to a tree creates a cycle.

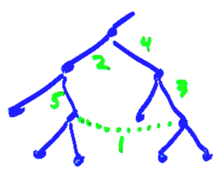
And deleting any cycle edge leaves a tree

Cor 1:

Solution to MST problem is a Tree.

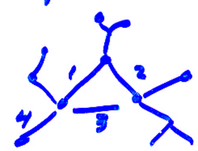
Cor 2:

Cheapest edge in E is in T.



Exercise: show 2^{nd}-cheapest $\in T$, too.

3^{rd} cheapest? Maybe not:



An Algorithm: Kruskal's Algorithm

Sort edges.

$T \leftarrow \emptyset$

while $|T| < n-1$ do

  $e \leftarrow$ next cheapest edge

  if $T \cup \{e\}$ acyclic

    then $T \leftarrow T \cup \{e\}$

    else discard e.

Theorem: It works!

proof. By contradiction. Suppose greedy tree is ~~not~~ min.

- Suppose greedy picks edges $g_1 g_2 \cdots$
  suppose $g_1 \leq g_2 \leq g_3 \leq \cdots$ really $c(g_1) \leq c(g_2) \cdots$

- Let $e_1, e_2 \cdots$ be edges in some MST T
  suppose $e_1 \leq e_2 \leq \cdots$

- let k be least index s.t. edges g_k and e_k are not identical

- Suppose $e_1 e_2 \cdots$ maximizes k among all MST's.

- Claim1: $T \cup \{g_k\}$ has a unique cycle & it (the cycle) contains at least one edge f s.t.
  (1) $f \notin \{g_1, g_2, \cdots\}$ and
  (2) $c(f) \geq c(g_k)$

- claim2: $T - \{f\} \cup \{g_k\}$
  (1) is a spanning tree
  (2) costs no more than T
  (3) has larger "k"

  A contradiction QED

*Proof: $\{g_1 \cdots\}$ is a tree, so contains no cycle, so some cycle edge must be $\notin \{g_1 \cdots\}$

Proof: Let f be any cycle edge $\notin \{g_1, \cdots\}$. If $c(f) < c(g_k)$, greedy alg. will consider f before g_k. But then it would have selected it, since $\{g_1, g_2, \cdots, g_{k-1}\} \cup \{f\} \subseteq T$, hence acyclic. This is a contradiction, so $c(f) \geq c(g_k)$