OH Today: 1:15 pm - 2:15 pm.

Exam 2: Tue, Apr. 6.

Minimum Spanning Trees

Input: Undirected, connected graph $G = (V, E)$ with non-negative edge weights $(positive)$

Output: Minimum Spanning Tree of $G$.

Algorithms

Kruskal's alg

- greedily process edges in $\mathcal{O}(\log E)$ and add them as long as no cycle is created.

Prim's alg
- "Wrong Dijkstra"
  - Choose the smallest cut edge that crosses the cut \((S, V \setminus S)\)

Reversi Delete
- Process edges in a order of their weights and delete an edge if removing it does not disconnect the graph.

**Lemma**: Let \( S \subset V \), \( S \neq \emptyset \), be a subset of vertices. Let \( e = (u,v) \) be the minimum wt. edge that crosses the cut \((S, V \setminus S)\). Then \( e \) belongs to every MST.
Corollary

**Theorem**: Kruskal's alg. works.

**Proof**: We will prove the following.

- Every edge added by Kruskal also belongs to "God's soln." (optimal soln.).
- Our soln. is a feasible soln., i.e., our soln. yields a Spanning Tree.

Let \( e = (u, v) \) be an edge that is about to be added to the soln. by Kruskal.

Why is \( e \) added to the solution?

Adding \( e \) does not create a cycle.

To show that \( e \) belongs to an optimal
8th, we will show that $e$ is the edge with smallest cut that crosses some cut $(S, V\setminus S)$.

$S$: connected component containing $u$

Clearly, $v \notin S$.

$e$ must be the edge with smallest cut
cross the cut \((S, V \setminus S)\). If there was some other edge \(f\) with a smaller cut than \(e\) s.t. \(f\) crosses \((S, V \setminus S)\), we would proceed \(f\) before \(e\), a contradiction.

Our solution is feasible. Acyclic is easy as Kruskal's alg. makes sure of that. If the resulting graph is not connected then we consider the cut that has one conn. comp on one side & the rest of the vertices on the
other. Since $G$ is connected, some edge must cross this cut $A$ or $A^c$. Thus, there would have included one of the edges that cross this cut, a contradiction.
Theorem: Prim's algo works.

Proof: $e = (u, v)$: edge about to be added by Prim.

$S$ from the algo. is the "S-side of the cut". That is, all vertices in $S$ from one side of the cut. Done. $S$

What remains to show is that the old is conn. & acyclic.
Acyclic is easy as we always have a new vertex if. Connected proof is same as before.

Lemma: Let $C$ be a cycle in $G$. Let $e = (u,v)$ be the heaviest weight edge in $C$, then no MST contains $e$.

Proof: Assume for contradiction that some MST, $T$, contains $e$.

Consider $T' = T - e$. 
$T'$ has exactly 2 cc. Follow the $uv$ path along the "other side" of the cycle & let $f$ be the first edge in $C$ that crosses the cut.

$T' + f$ gives a spanning tree of a smaller cut, a contradiction.

Thus: Reverse delete alg. works.

Proof idea: T-S-T: any edge that $w$
removal must also be removed by "God".

- $e = (u,v)$ is about to be removed by ourselves.

- Removing $e$ does not disconnect the graph, i.e., after removing $e$, there is a $u,v$ path.

- The $u,v$ path along with $e$ forms a cycle in $G$.

- Since edges proceed in a order of $e$, $e$ must be the heaviest edge in the cycle.
apply the lemma.

What if the edge weights are not distinct?

\[ \Lambda = \begin{array}{c}
3, 3, 3, 5, 5, 8, 20, 21, 21, 21, 25, \\
8, 3.2, 3.3, 5.1, 5.2, 8.1, 20.1, 21.1, \ldots \\
3 + \epsilon, 3 + 2\epsilon, 3 + 3\epsilon \ldots \end{array} \]

\[ e = \frac{\Lambda}{n^2} \text{ smallest "gap"} \]

**Input:** DAG \( G = (V, E) \)

\( s \in V \)

cuts on edges (positive)

**Obj:** To find shortest paths from
8 to every vertex in $G$. in $O(n+m)$ time.

1. Topological Sort $G$. $O(n+m)$. 

\[
\begin{align*}
& a & b & c & d \\
\end{align*}
\]
for each $u \in V$ do

$d[u] \leftarrow \infty$

$\pi[u] \leftarrow \text{NIL}$

$d[s] \leftarrow 0$

$L \leftarrow \text{Topological Sort}(A)$

for each vertex $v$ in $L$ starting after $s$ do

\[
  d[v] \leftarrow \min \left\{ d[u] + w_{uv} \right\} \quad \text{for } u \in \text{in}(v)
\]

Set $\pi(v)$ accordingly.

$\pi(v) \in b.$
for each vertex \( u \) starting from \( u \) do

\[
\text{for each } v \in N(u) \text{ do}
\]

\[
\text{if } d[v] > d[u] + w_{uv} \text{ then}
\]

\[
d[v] \leftarrow d[u] + w_{uv}
\]

\[
\pi[v] \leftarrow u.
\]