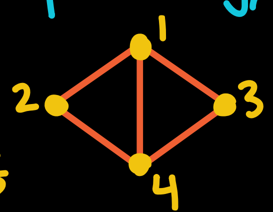
# Math 409: Discrete Optimization

## Today: Minimum Spanning Tree

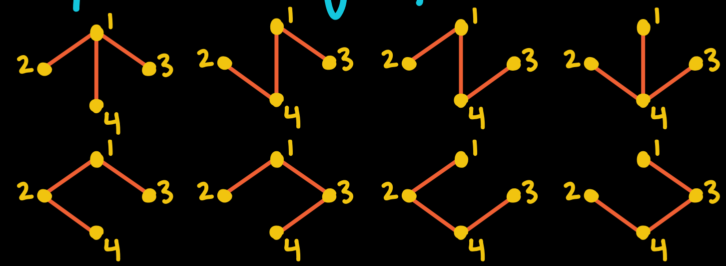
Recall: A <u>spanning tree</u> of a graph $G=(V,E)$ is a spanning, connected, acyclic subgraph of $G$.

Ex:



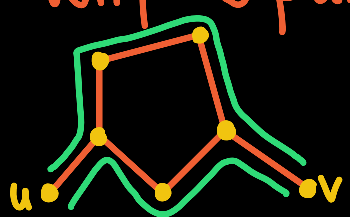

G

Spanning trees of G:




Equivalent: $T \subseteq E$ is a spanning tree of $G$ $\iff$ for all $u,v \in V$, there is a unique path $u \to v$ in $T$

Why? $\exists$ path $u \to v$ for all $u,v \in V$ $\iff$ connected & spanning




more than one path $u \to v$ (for some $u,v \in V$) $\iff$ contains a cycle

## MINIMUM SPANNING TREE

<u>Input</u>: graph $G=(V,E)$, edge costs $c_e \in \mathbb{R}_{\geq 0}$ for $e \in E$

<u>Goal</u>: a spanning tree $T$ of $G$ minimizing cost

$$C(T) = \sum_{e \in T} c_e$$

Cayley's formula: $\#\{\text{span. trees of } K_n\} = n^{n-2}$ ↙ comparing all costs infeasible for large n!

## Basic properties

**Thm:** Let $G=(V,E)$ be a connected graph with $|V|=n$.
Then (1) $|E| \geq n-1$

(2) $G$ has a spanning tree

(3) $G$ is acyclic $\iff |E|=n-1$

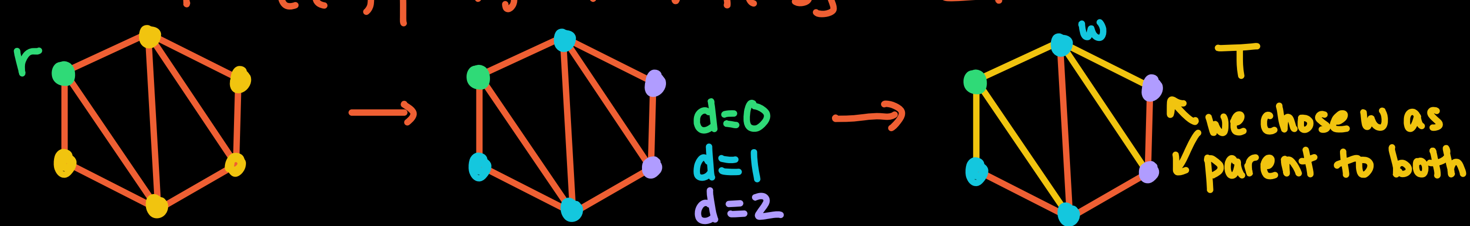
↳ $G$ is a spanning tree of itself

(Proof) Fix a vertex $r \in V$ to be the "root".
For each vertex $v \in V \setminus \{r\}$, define

$d(v)$ = min # edges of a path $v \to r$. ↙ path exists since $G$ is connected

$p(v)$ = some vertex $w$ so that $\{v,w\} \in E$ and $d(w) = d(v) - 1$ } might be more than one choice for this

Take $T = \{\{v, p(v)\} : v \in V \setminus \{r\}\} \subseteq E$.



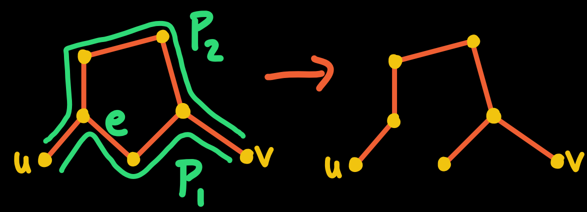

Observations: (1) $n-1 = |T| \leq |E|$

(2) $T$ is a spanning tree

For any $v \in V$, there is a path $v \to r$ in $T$ by taking parent nodes $v \to p(v) \to p(p(v)) \to \dots \to r$.

$\Rightarrow$ For $u, v \in V$, there is a path $u \to r \to v$.

If there are two paths $P_1, P_2$ from $u \to v$ in $T$ then for any edge $e \in P_1 \setminus P_2$, $(V, T \setminus \{e\})$ is still connected



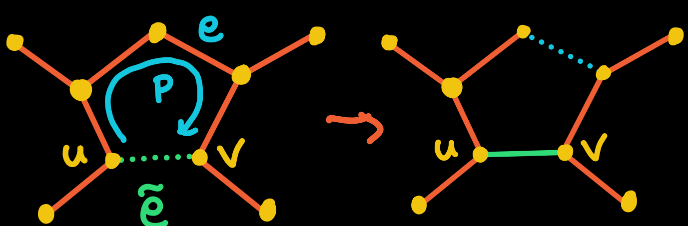

$\overset{(1)}{\Longrightarrow}$ $n-2 = |T \setminus \{e\}| \geq n-1$ ⚡

(3) $G$ is acyclic $\Leftrightarrow T = E \Leftrightarrow n-1 = |T| = |E|$

<u>Edge swaps</u>: to move around {span. trees of $G$}

This will let us move around {span. trees of $G$} and (hopefully) decrease costs

<u>Lemma</u>: Let $T \subseteq E$ be a spanning tree of $G = (V, E)$ and $\tilde{e} = \{u, v\} \in E \setminus T$. Let $P \subseteq T$ be the unique path $u \to v$ in $T$. Then for any edge $e \in P$ $\tilde{T} = (T \setminus \{e\}) \cup \{\tilde{e}\}$ is a spanning tree of $G$.




$T$ $\qquad$ $T \setminus \{e\} \cup \{\tilde{e}\}$

(Proof) $P \cup \{\tilde{e}\}$ is a cycle.

$\Rightarrow \tilde{T}$ still connected and $|\tilde{T}| = |T| = |V| - 1 \Rightarrow$ span. tree

Idea for min cost spanning tree: start with some sp. tree and swap edges to lower cost

$$C(\tilde{T}) = C(T) - c_e + c_{\tilde{e}}$$

**Thm**: Let $T$ be a spanning tree of $G=(V,E)$ and $c_e \in \mathbb{R}_{\geq 0}$ for all $e \in E$. Then $T$ is a minimum cost spanning tree (MST)
$\iff$ no edge swap for $T$ decreases the cost.

Edge swap: $T \to \tilde{T} = T \setminus \{e\} \cup \{\tilde{e}\}$ s.t. $\tilde{T}$ is a sp. tree of $G$

($\Rightarrow$) Clear  ($\Leftarrow$) Proof next time

**Significance**: For spanning trees, local optimality (no "small" improving moves) implies global optimality (lowest cost among all possible)
This makes finding MST easier than many other problems in discrete optimization.