Math 409: Discrete Optimization

Today: Minimum Spanning Tree (cont')

MINIMUM SPANNING TREE

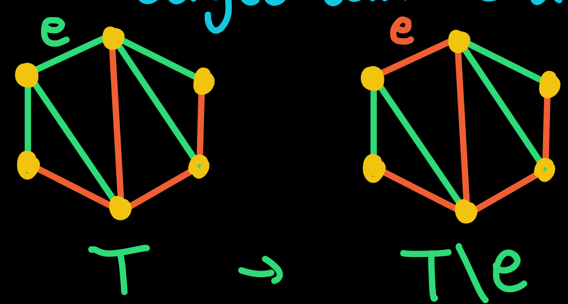
<u>Input</u>: graph $G=(V,E)$, edge costs $c_e \in \mathbb{R}_{\geq 0}$ for $e \in E$

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

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

<u>Local moves in $\{$sp. trees of $G\}$</u>: edge swaps

Given a spanning tree T of G and edge $e \in T$ what edges can we add to $T \setminus \{e\}$ to get a sp. tree?

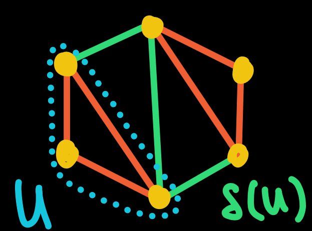


Note: $T \setminus \{e\}$ is disconnected (it only has $|V|-2$ edges)

Need to reconnect

For $U \subseteq V$, define $\delta(U) = \{e \in E : |e \cap U| = 1\}$

This is a <u>cut</u> of G.



Check!: If U is a connected component of $T \setminus \{e\}$, then $T \setminus \{e\} \cup \{\tilde{e}\}$ is a sp. tree of G
$\iff \tilde{e} \in \delta(U)$

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)
$\Leftrightarrow$ 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

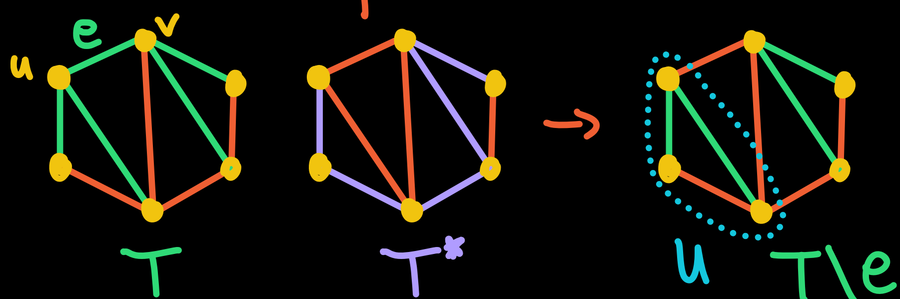
($\Leftarrow$) (By contrapositive) Take

T = sp. tree of G, not minimal cost

T^* = min cost sp. tree (with maximal $|T \cap T^*|$)

Take $e=\{u,v\} \in T \setminus T^*$ (possible since T not min cost $\Rightarrow T \neq T^*$)

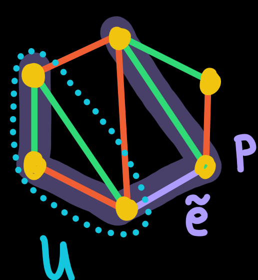
and $\mathcal{U}$ = connected component of u in $T \setminus \{e\}$



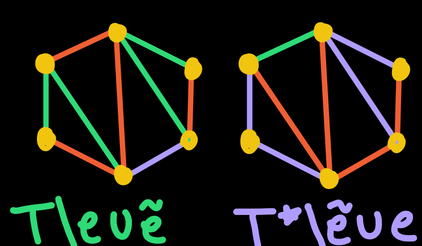
Let P = unique path $u \to v$ in T^*.

Since $u \in \mathcal{U}$, $v \in V \setminus \mathcal{U}$, $\delta(\mathcal{U}) \cap P$ nonempty.

Take $\tilde{e} \in \delta(\mathcal{U}) \cap P \subseteq \delta(\mathcal{U}) \cap T^*$.



Both $T \setminus \{e\} \cup \{\tilde{e}\}$ and $T^* \setminus \{\tilde{e}\} \cup \{e\}$ are spanning trees of G



$c(\tilde{e}) > c(e) \Rightarrow T^* \setminus \{\tilde{e}\} \cup \{e\}$ has lower cost than T^*

$c(\tilde{e}) = c(e) \Rightarrow T^* \setminus \{\tilde{e}\} \cup \{e\}$ has same cost as T^* but more edges in common with T

Both contradict choice of T^*. Only possibility:

$c(\tilde{e}) < c(e) \Rightarrow$ edge swap $T \to T \setminus \{e\} \cup \{\tilde{e}\}$ decreases cost

KRUSKAL'S ALGORITHM

Input: a connected graph $G = (V, E)$, edge costs $c : E \to \mathbb{R}$

Output: a min. cost spanning tree of G

(1) Sort edges by cost: $c_{e_1} \leq \ldots \leq c_{e_m}$ $\quad (m = |E|)$

(2) Initiate $T_0 = \emptyset$

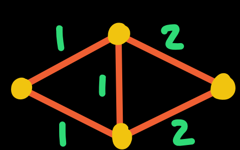
(3) For $i = 1, .., m$ do

if $T_{i-1} \cup \{e_i\}$ is acyclic, $T_i := T_{i-1} \cup \{e_i\}$

else $T_i := T_{i-1}$

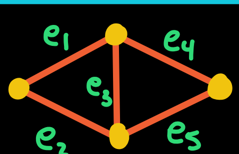
(4) Output $T = T_m$

Ex:



costs

$\rightarrow$



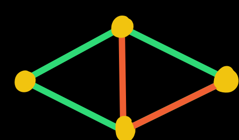
sorted edges

$T_0 = \{\}$

$T_1 = \{e_1\}$

$T_2 = \{e_1, e_2\} = T_3$

$T_4 = \{e_1, e_2, e_4\} = T_5 = T$



T

<u>Thm</u>: Output T is a min cost sp. tree of G

(Proof) <u>Claim 1</u>: T is a sp. tree of G.

Let C be a connected comp. of (V,T).

Suppose $C \neq V$. Since G is connected, $\delta(C) \neq \phi$

For any $e_i \in \delta(C)$, $T \cup \{e_i\}$ is acyclic.

$T \cup \{e_i\}$ is acyclic $\Rightarrow T_{i-1} \cup \{e_i\}$ acyclic $\Rightarrow e_i \in T_i \subseteq T$

Contradicts $e_i \in \delta(C)$ for some conn. comp. C of $T \Rightarrow C = V$

<u>Claim 2</u>: T has min cost.

If not, for some $e_j \in T$, $e_i \notin T$,

$T \setminus \{e_j\} \cup \{e_i\}$ gives lower cost sp. tree (by thm)

$\Longrightarrow c(e_j) > c(e_i) \Rightarrow j > i$

$T \setminus \{e_j\} \cup \{e_i\}$ a sp. tree $\Rightarrow$ acyclic

Since $i < j$, $T_{i-1} \subseteq T \setminus \{e_j\} \Rightarrow T_{i-1} \cup \{e_i\}$ acyclic $\Rightarrow e_i \in T$ $\square$

Running time?

(1) Sorting m values can be done in time $O(m \log(m))$

(3) $T_{i-1} \cup \{e_i\}$ has a cycle $\Longleftrightarrow$ end pts $\{u,v\} = e_i$ belong to same conn. component of T_{i-1} (checked in time $O(\log(n))$)

Doing this m times takes $O(m \log(n))$