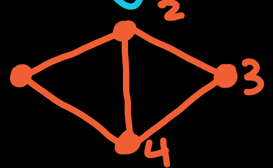
MA/AMA 514:
Networks & Combinatorial Optimization

Today: §1.4 Minimum spanning trees

Graphs - many definitions!

An undirected graph (or just graph) is a pair $G=(V,E)$ of
- a set of vertices V
- a set of edges E

"edge" = unordered pair of vertices

Ex: 

$V=\{1,2,3,4\}$ $E=\{12,14,23,24,34\}$

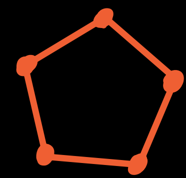
A walk in G is a sequence $v_0, e_1, v_1, e_2, \ldots, e_m, v_m$ where $v_j \in V$, $e_j=\{v_{j-1}, v_j\} \in E$

A path is a walk with distinct vertices $v_0, \ldots, v_m$

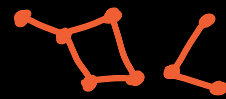
G is connected if there is path between any two vertices of G

A cycle in G is a walk with distinct edges and $v_0=v_m$

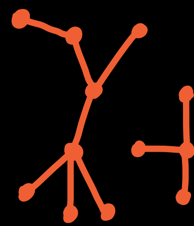
A <u>circuit</u> in G is a cycle with all vertices distinct except $v_0 = v_m$



The <u>(connected) components</u> of G are the maximal connected subgraphs.

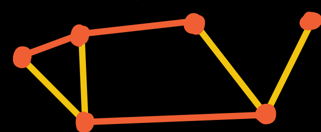


A graph with no cycles is a <u>forest</u> and a connected forest is a <u>tree</u>



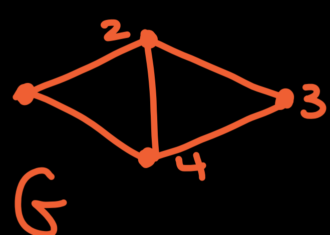
Check: every forest in a disjoint union of trees!

A subgraph $H = (V', E')$ of G is <u>spanning</u> if $V = V'$ and every vertex $v \in V$ is contained in some edge $e \in E'$.



A <u>spanning tree</u> of G is a spanning, connected subgraph of G with no cycles.

Ex.



$T = \{12, 24, 23\}$
$\tilde{T} = \{14, 24, 23\}$ both sp. trees of G

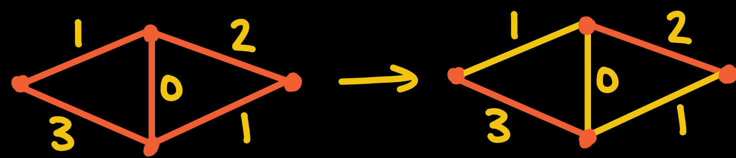
Minimum Spanning Tree (MST) Problem

<u>Input</u>: a connected graph $G = (V, E)$ and a "length" function $\ell : E \to \mathbb{R}$

<u>Goal</u>: find a min. length spanning tree, i.e. a sp. tree T minimizing

$$\ell(T) = \sum_{e \in T} \ell(e)$$

Cayley's formula: complete graph $K_n = ([n], \binom{[n]}{2})$ has n^{n-2} spanning trees!

Ex: 

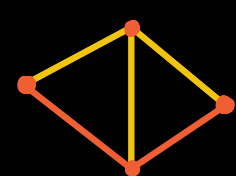
min. spanning tree has length 2

Applications: designing road systems, electrical grids, telephone lines, etc.

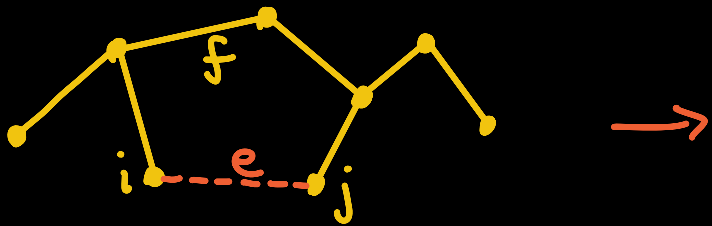
Useful facts about spanning trees (You check!)

1) A spanning, connected subgraph H of G is a spanning tree if and only if $|E(H)| = |V| - 1$

2) In a spanning tree, there is a unique path between any two vertices.



3) If $e = \{i,j\} \notin T$ and $f \in T$ lies on the unique path in T between i and j, then $T \cup \{e\} \setminus \{f\}$ is a spanning tree of G.



Dijkstra-Prim Algorithm

(1959, 1957, based on Borůvka (1926))

1) Choose any vertex $v \in V$. Set $V_0 = \{v\}$, $E_0 = \phi$

2) While $V_k \neq V$, choose $e_k \in \delta(V_k)$ of min. length
   Set $V_{k+1} = V_k \cup e_k$ and $E_{k+1} = E_k \cup \{e_k\}$.

3) Output (V_k, E_k) $\quad (k = |V|-1)$

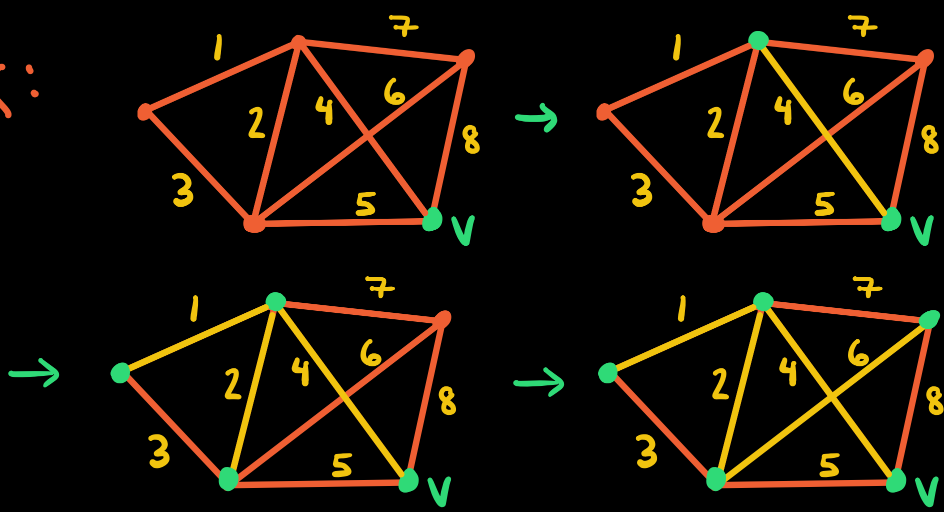
Here $\delta(S) = \{e \in E$ with one end pt. in S, one in $V \setminus S\}$

↑ Example of a greedy algorithm!

\# steps $= |V|$ $\quad$ Running time $\leq |V| \cdot |E|$ (At every step, compare $\leq |E|$ edges)

Thm 1.12: $O(|E| + |V| \log(|V|))$

Ex:



min spanning tree

$\ell(T) = 1 + 2 + 4 + 6 = 13$

Proof of correctness:

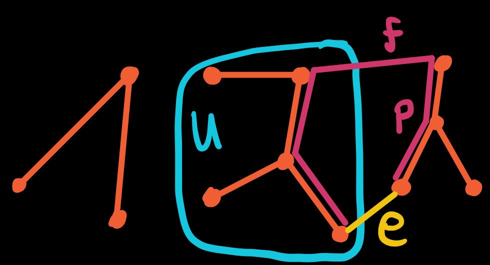
Call a forest of G <u>greedy</u> if it is contained in a minimum spanning tree of G.

Thm (1.11) Suppose

- F is a greedy forest of G
- U is the vertex set of a conn. component of F
- $e \in \delta(U)$ has min. length in $\delta(U)$

then $F \cup \{e\}$ is a greedy forest of G.

(Proof) F greedy $\Rightarrow \exists$ min. spanning tree $T \supseteq F$



If $e \in T \rightarrow$ done!

Otherwise, take

P = unique path in T between end pts. of e

$\Rightarrow \exists f \in E(P) \cap \delta(U)$

By useful fact (3), $T' = T \cup \{e\} \setminus \{f\}$ also a spanning tree of G.

Since $\ell(e) \leq \ell(f)$, $\ell(T') \leq \ell(T)$

$\Rightarrow T'$ is a min. spanning tree of G.
$f \notin F \Rightarrow F \cup \{e\} \subseteq T' \Rightarrow F \cup \{e\}$ greedy forest

Cor: The Dijkstra-Prim alg. gives a minimum spanning tree of G

(Proof) Start: $F = (\{v\}, \emptyset)$ is a greedy forest
By Thm, every update (V_k, E_k) is also.
$\Rightarrow$ Output = a spanning, connected greedy forest
= a min. spanning tree

Another option for finding a min. spanning tree:

Kruskal's Algorithm (1956)

1) Sort edges s.t. $\ell(e_1) \le \ell(e_2) \le \ldots \le \ell(e_m)$ $(|E| = m)$
2) Set $F_0 = (V, \emptyset)$
3) For $i = 1, \ldots, m$, if $F_{i-1} \cup \{e_i\}$ acyclic, $F_i = F_{i-1} \cup \{e_i\}$
otherwise $F_i = F_{i-1}$.

Output $T = F_m$.

Run time: $O(|E| \log(|E|))$
sorting edges!

(Proof of correctness) Start F_0 is a greedy forest

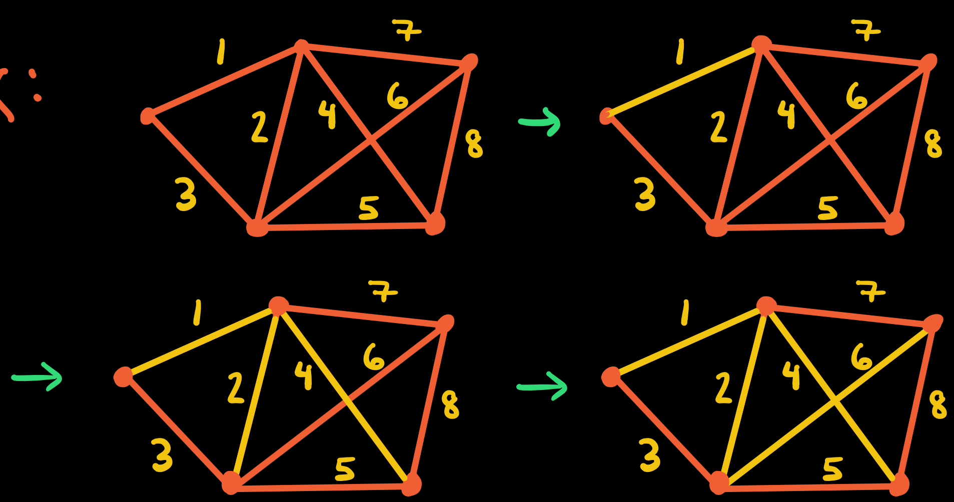
If $F_{i-1} \cup \{e_i\}$ is acyclic, e_i connects two connected components $\mathcal{U}, \mathcal{U}'$ of $F_{i-1} \Rightarrow e_i \in \delta(\mathcal{U})$.

If $\ell(f) < \ell(e_i)$, end pts of f are connected in F_{i-1}.

$\Rightarrow e_i$ has min length among $\delta(\mathcal{U})$.

$\Rightarrow$ By Thm 1.1, F_i is a greedy forest.

Ex:



min spanning tree

$\ell(T) = 1 + 2 + 4 + 6 = 13$