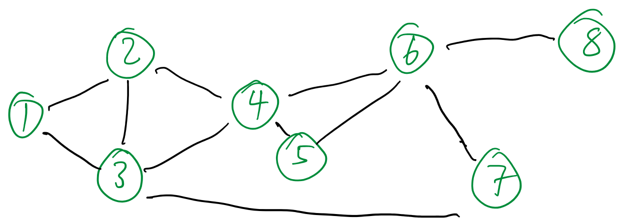
Lec01-MST

Sunday, August 25, 2024 10:48 PM

KT 3.1, 4.5, 4.7

## Minimum Spanning Trees (MST)

**Problem:** cost-effective wiring of a network




locations

possible connections

**Other applications:** (1) DNA sequences & similarity

└ recapitulate evolutionary tree

(2) Clustering by removing long edges

**Graphs** An undirected graph $G=(V,E)$ is a pair of sets

$V$ = set of vertices / nodes

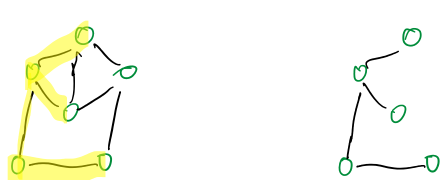
$E$ = set of 2-element subsets of $V$

$e \in E \implies e = \{u,v\}$ with $u,v \in V$.

A graph is directed if $E$ is a set of ordered pairs $(u,v)$, $u,v \in V$.

A graph $H=(V_H, E_H)$ is a subgraph of $G=(V_G, E_G)$ if

$V_H \subseteq V$ and $E_H \subseteq E$



A connected component is a maximal connected subgraph

A connected component is a maximal connected subgraph

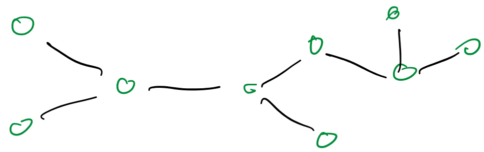
Graphs naturally model many concepts

↳ anything that has objects + relationships b/t pairs of objects

1. social networks
2. geographic adjacency
3. polyhedra
4. chemical molecules
5. assigning jobs to applicants
6. food webs
7. finite-state machines
8. markov processes
9. project dependencies
10. WWW
11. telephone networks
12. roads
13. neural networks
14. phylogenetic relationships
15. mesh approximations to surfaces

Def. A Cycle of a graph $G=(V,E)$ is a sequence of distinct vertices $v_1, \dots, v_k \in V$ s.t. $\{v_i, v_{i+1}\} \in E$ $\forall\, i=1,\dots,k-1$ and $\{v_k, v_1\} \in E$

Def. A tree is a graph that is connected + has no cycles.



## MST problem

Given an undirected, connected graph $G$, and nonnegative weights $d(u,v)$ = cost of using edge $\{u,v\}$,

Find the subgraph $T$ that connects all vertices & minimizes

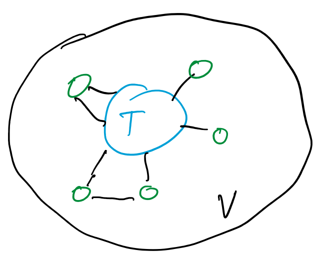
$$\text{cost}(T) = \sum_{\{u,v\} \in T} d(u,v)$$

Why will $T$ be a tree? If not, ...

## Prim's alg

- Given graph $G=(V,E)$, select arbitrary node $s$ as start of $T$
- Repeat $|V|-1$ times:

  Add to $T$ the lowest cost edge $\{u,v\}$ where $u \in T$ & $v \notin T$.




## Questions:

Correctness

Data structure / implementation - how to quickly find next lowest cost edge

Worst-case runtime

Is it possible to be faster / complexity

## Proof of correctness

**Theorem** (characterization of trees)

The following are equivalent:

1. T is a tree.
2. T contains no cycles and $n-1$ edges
3. T is connected and has $n-1$ edges
4. T is connected and removing any edge disconnects it
5. Any two nodes in T are connected by exactly one path
6. T is acyclic, and adding any new edge creates exactly one cycle

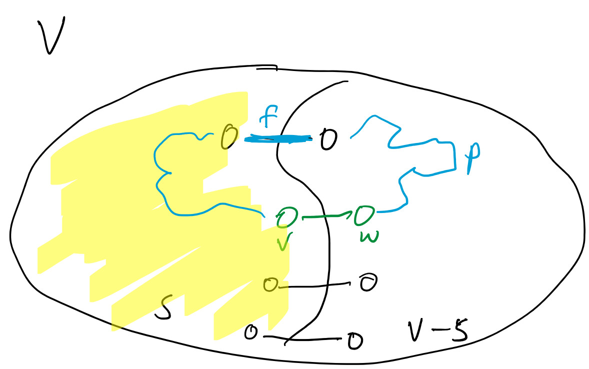
good exercises to verify

**Assumption** No two edges in G has the same cost

else, can add small random $\varepsilon_e$ to weight of every edge e.

**Theorem** (MST Cut Property)

Let S be a subset of nodes with $|S| \geq 1$ and $|S| < |V|$.
Every MST contains the edge $e = \{v, w\}$ with $v \in S$ and $w \in V-S$ of min weight




A pair $(S, V-S)$ is a cut of the graph

**proof.** Suppose $e \notin T$. Then because T is connected, it must contain a path P
between v and w ...

proof. Suppose $e \notin T$. Then because $T$ is connected, it must contain a path $P$ between $v$ and $w$, and $P$ must contain an edge $f$ that crosses the cut.
Then subgraph $T' = (T - f) \cup e$ has lower weight than $T$.
$T'$ is acyclic because the only cycle in $T' \cup f$ is eliminated by removing $f$.

Thm (Prim's correctness) At termination, Prim's always returns a MST.

proof. At any point, $T = (V_T, E_T)$ is a subgraph that is a tree.
$T$ grows by 1 vertex & 1 edge at each step, so it will stop after $|V_G| - 1$ steps, and will be a spanning tree.
The pair $(V_T, V_G - V_T)$ is a cut of $G$.
By the cut property, the MST contains the lowest cost edge crossing this cut, but that is exactly the next edge Prim's always adds.

So, Prim's only adds edges in the MST.

$\boxtimes$