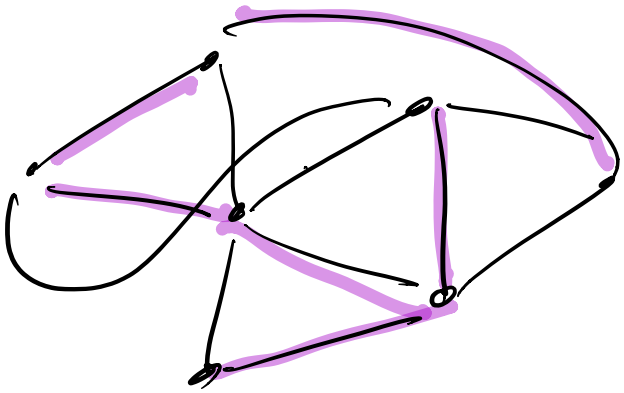
General stuff about trees

Plot: looking for spanning subtrees in connected graphs



Proposition Suppose H is a spanning subgraph of G. Then the following are equivalent:

1) H is a tree
2) # of edges in H = (# vertices in H) = (# vertices in G) -1
3) H is connected and no proper spanning subgraph of H is connected (H is minimal connected)
4) H is acyclic and no subgraph H' of G containing H properly is acyclic (H is maximal acyclic)

Why? (showed $1 \Rightarrow 2$)

lets assume

$1 \Leftrightarrow 4$

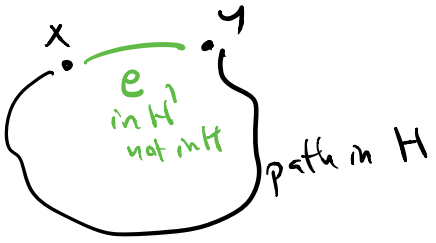
$\Rightarrow$

Suppose $H \subset G$ is a tree, know H acyclic.

if $H' \supsetneq H$, why is H' not acyclic.

G

if e is an edge in H' not in H, say connecting x & y then since H was already spanning & connected, $\exists$ path from x to y in H. Following with e, get a cycle in H'



Conversely ($\Leftarrow$) if H is maximal acyclic

why is it a tree?

need to show connected:

pick $x, y \in G$ (vertices same in G & H)

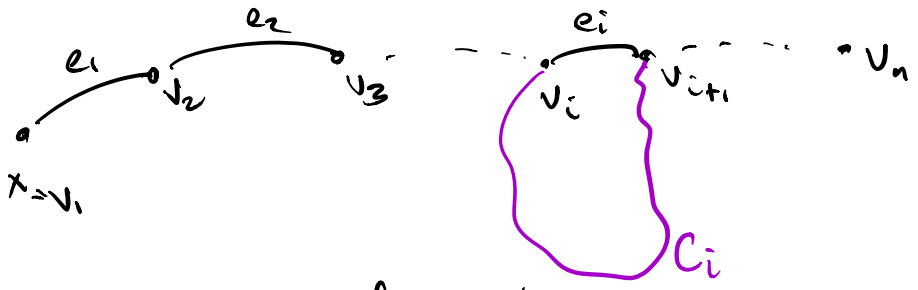
since G is connected, can find a walk

$x = v_1$ $y = v_n$ walk: $v_1 e_1 v_2 e_2 \cdots e_{n-1} v_n$

for each i, either $e_i \in H$ or if not

$H + e_i$ has a cycle involving e_i

C_i



define $P_i = \begin{cases} e_i \text{ if } e_i \in H \\ C_i - e_i \text{ if } e_i \notin H \end{cases}$

then $P_1 P_2 \cdots P_{n-1}$ is a path from v_1 to v_n

Kruskal's Algorithm

Start w/ a graph G which is connected, with weights assigned to each edge

$$G = (V, E) \qquad \omega : E \longrightarrow \mathbb{R}_{\geq 0}$$

Problem: Find a spanning tree $T \subset G$ such that the sum of weights of edges in T is as small as possible.

Kruskal:

at each step, add an edge to our subgraph
which one? smallest weight which doesn't create a cycle.

Amazingly, this always gives a tree with minimum possible total weight.

Since result is maximal (w.r.t. to edges) acyclic, it is a tree.

Optimal?

Imagine our construction adds edges in the sequence

$e_1, e_2, \ldots, e_{n-1}$ giving $T \subset G$

consider all minimal spanning trees

let H be one with edges $e_1, \ldots, e_k$ k large as possible.

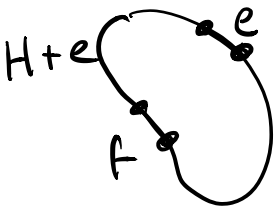
let $e = e_{k+1}$. H max'l acyclic $\Rightarrow$ $H + e$ has a cycle

and some edge f in this cycle is not in T

H f e

consider $H + e - f$ is connected

and same # of vertices & edges as G so a tree.



H minimal $\Rightarrow$ so $H+e-f$ is no lighter than H

$$w(H+e-f) = w(H) + w(e) - w(f) \geq w(H)$$

$$\Rightarrow w(e) \geq w(f)$$

but $e_1, e_2, \ldots, e_k, f$ is still acyclic.

adding edges $\ldots$ since these are in H which is acyclic.

so we could have added f instead of $e = e_{k+1}$ in algorithm.

$$\Rightarrow w(e) \leq w(f) \quad \text{(since added } e\text{)}$$

$$\Rightarrow w(e) = w(f)$$

$\Rightarrow H+e-f$ has same wt as H

but one more edge in common w/ T ($e_{k+1} = e$)

it has

□

Let $\mathcal{C} = \{$ all $HCG \mid H$ spanning subtree $\}$

$$w: \mathcal{C} \longrightarrow \mathbb{R}_{\geq 0}$$

$$H \longmapsto w(H)$$

let w_0 = smallest value attained by this.

Let $\mathcal{T}_0 = \{ H \subseteq G \mid H \text{ spanning subtree } \& \; w(H) = \omega_0 \}$

agree: $\mathcal{T} \longrightarrow \mathbb{Z}_{\geq 0}$

$H \longmapsto \{ \max j \text{ s.t. } e_1, \ldots, e_j \in H \}$

Let k = max value of agree on an elmt of $\mathcal{T}_0$

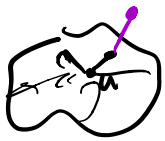
Let $H \in \mathcal{T}_0$ s.t. agree$(H) = k$.

we showed that if $k \neq n-1$, then found an edge f in H w/ $w(f) = w(e_{k+1})$ s.t. $H - f + e_{k+1}$ is also a spanning tree.

Prim's algorithm

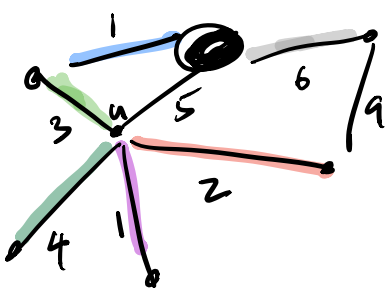
Start w/ an arbitrary vertex u.

at each step, add a leaf from what we've constructed so far of minimal weight.



Inductively construct a subgraph $T \subset G$

each step, add an edge of minimal length such that only one vertex of edge is in T (and add other vertex)



step #s

1 2 3

4 5 6