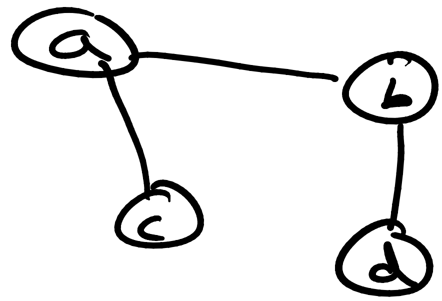
Lecture 6 - Trees

Monday, January 30, 2023 12:58 PM

Review: Havel-Hakimi

$$S = \{\overset{a}{\cancel{2}}, \overset{b}{2}, \overset{c}{1}, \overset{d}{1}\}$$



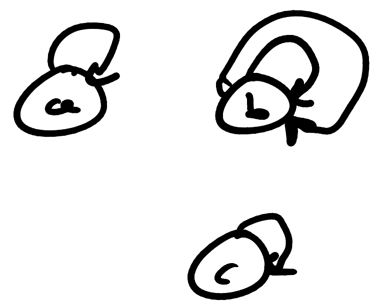
$$S' = \{\overset{-1}{\overset{b}{1}}, \overset{-1}{\overset{d}{1}}, \overset{c}{0}\}$$

$$S'' = \{0, 0\}$$

For directed graphs

$$S = \{\overset{a}{(1,1)}, \overset{b}{(2,2)}, \overset{c}{(1,1)}\}$$

$$\begin{array}{cccc} a & b & b & c \\ \downarrow & \downarrow & \downarrow & \downarrow \\ a & b & b & c \end{array}$$



Bag o' tricks

Structural arguments

- consider v of degree 2, etc.

Extremal arguments

Extrenal arguments

- consider max path $P \subseteq G$

Parity arguments

- even + even = even, odd + odd = even
  odd + even = odd

Weak induction

construct

$P(1) \ldots P(k), P(k+1)$

basis assme show

Strong induction

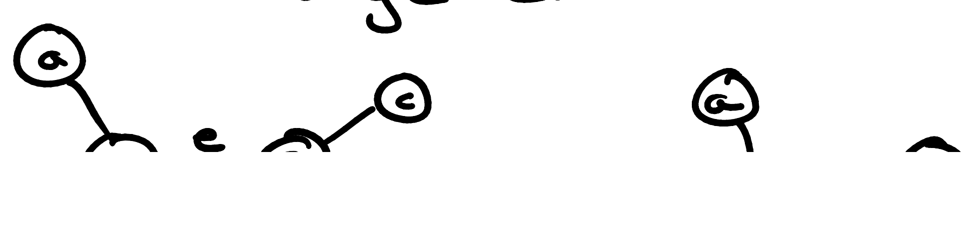
construct

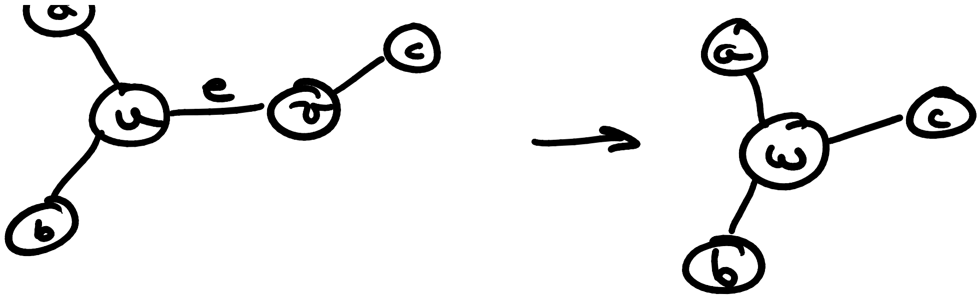
$P(1) \ldots P(k) \ldots P(n)$

basis assme show

Constructions:

- vertex addition / deletion
- edge addition / deletion
- subgraph deletion
- edge contraction





Necessity & sufficiency

- equivalence (<=>)
- to prove: prove one direction (=>) then the other (<=)

Proof by algorithm

- Here's an explicit algorithm to prove property P

Proof by counter-example

- can we show X?
  No, here's a counter-example

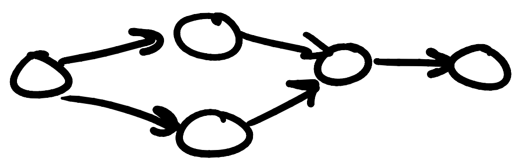
Trees 

Tree: a connected undirected simple acyclic graph simple acyclic graph

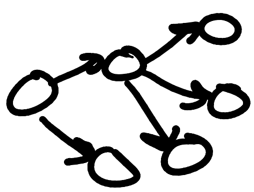
↳ contains no cycles

Forest: an undirected simple acyclic graph

DAG: directed acyclic graph



Polytree: a DAG where the underlying graph is a tree



↳ undirected graph if we ignore edge directivity

Tree T necessary conditions

T is minimally connected

→ $T - e$ is disconnected $\forall e \in E(T)$

T is maximally acyclic

→ $T + e$ creates a cycle

$\rightarrow T+e$ creates a cycle
$e=(u,v),\ u,v \in V(T)$

T has $|E(T)| = |V(T)| - 1$

T has a single unique
u,v-path $\forall u,v \in V(T)$

T is bipartite

Braincercize: which of the above
properties are also sufficient?

Prove: T is a tree $\rightarrow T$ is bipartite

weak induction on $m = |E(T)|$
$(n = |V(T)|)$

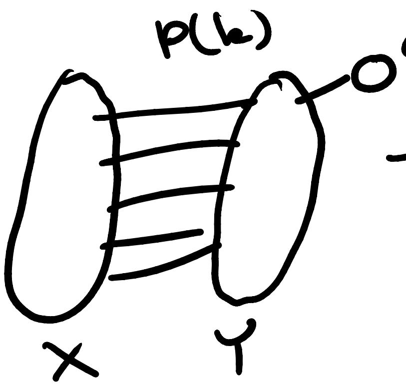
Basis $P(0) \rightarrow \circ$ ✓

Assume we have $P(k)$ tree,
and via I.H. $P(k)$ is bipartite

Add an edge (+ new vertex) to
create $P(k+1)$

Note: we already have a valid bipartitioning for P(k)

Note 2: P(k+1) adds a new leaf



→ we can place the new leaf in the opposite set as its neighbor

⇒ P(k+1) is bipartite

Prove: T is tree → $\forall u,v \in V(T)$: $\exists$ unique u,v-path

strong induction on $m = |E(T)|$

Basis P(0) → 0 ✓

consider tree P(n)

↳ construct P(k) s.t. P(k) is a tree

Note: all trees have leaves

Let's remove a leaf edge to construct P(k) construct $P(k)$

As removing a leaf from a tree doesn't result in a cycle, $P(k)$ is guaranteed to be a tree

→ I.H. on $P(k)$ gives us unique u,v-paths $\forall u,v \in V(P(k))$

Bring it on back to $P(n)$

- by adding back edge e and leaf l

Bring it on home

- showing our result via I.H. still holds on $P(n)$

We already know

$\{\forall u, v \in V(P(n)) : u, v \neq l\} : \exists$ unique u,v-path

To get the rest of u,l-paths

→ we simply add edge e to unique u,x-paths, where x is l's neighbor

u, x-paths, where x is ℓ's neighbor $\square$

Extra fun definitions

distance: $d(u,v)$ = length of shortest u,v-path

diameter: $D(G)$ = length of the longest u,v-path over $\forall u,v \in V(G)$

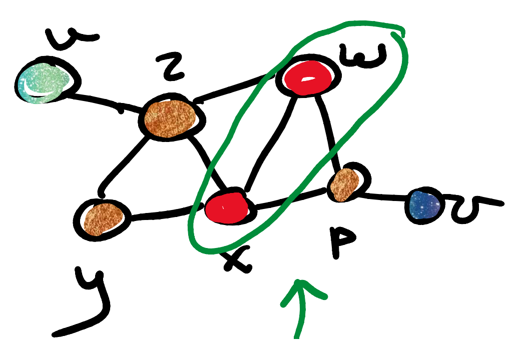
$$D(G) = \max_{\forall u,v \in V(G)} d(u,v)$$

eccentricity: $\epsilon(v) = \max_{\forall u \in V(G)} d(u,v)$

radius: $R(G) = \min_{\forall v \in V(G)} \epsilon(v)$

center of G = the induced subgraph of vertices with minimum eccentricity in G

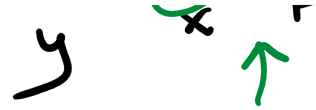
G



$d(u,v) = 4$

$D(G) = 4$

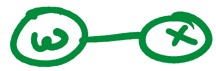
$\epsilon(w) = 2$

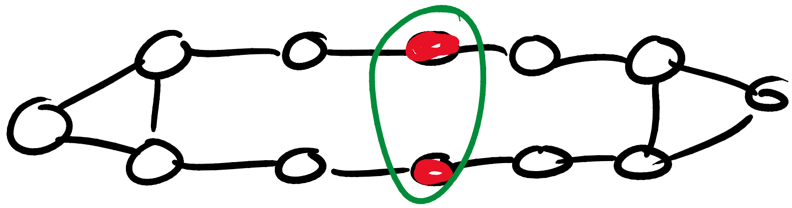


center

$e(w) = 2$

$R(G) = 2$





center