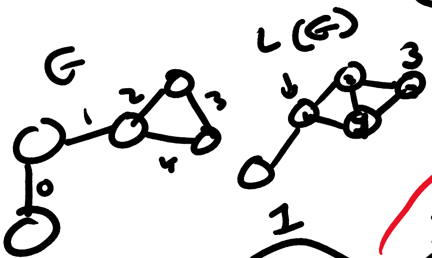
Lecture 24 - Line Graphs and Edge Coloring

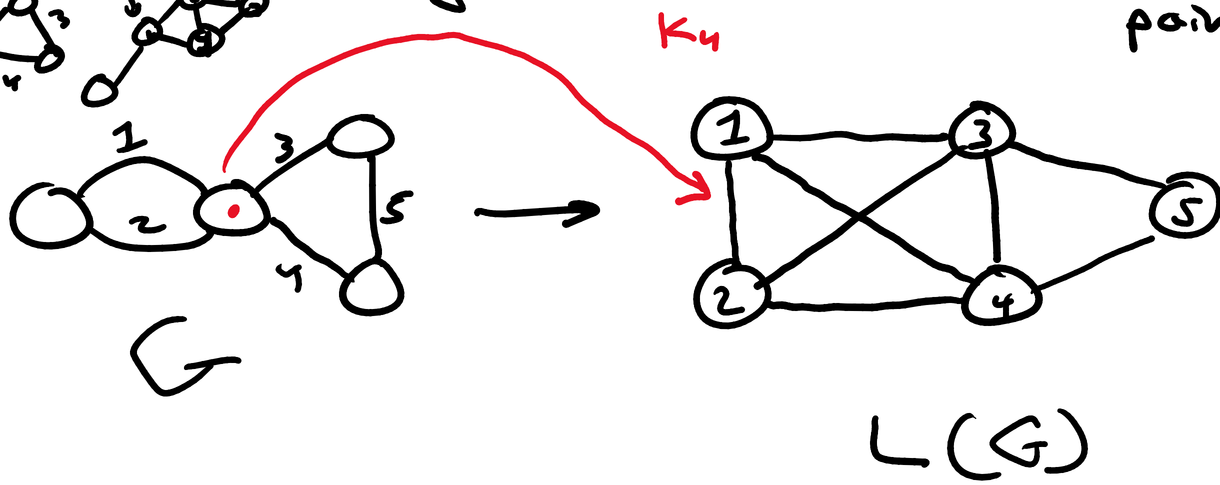
Monday, April 18, 2022 6:06 AM

Line Graphs

The line graph of $G \rightarrow L(G)$

defined $\begin{cases} \text{edges of } G \rightarrow \text{vertices of } L(G) \\ \text{edges in } L(G) \text{ exist where} \end{cases}$
edges in G share an end point





Note: the equivalence of
edges of $G \leftrightarrow$ vertices of $L(G)$
is relevant to several problems:

1. Euler Tour on $G \leftrightarrow$ spanning cycle on $L(G)$

2. Matching on $G \leftrightarrow$ independent set on $L(G)$

3. Cut edge on $G \leftrightarrow$ cut vertex on $L(G)$
   $e = (u, v)$ if $d(u) > 1$
   $d(v) > 1$

$e = (u, v)$ if $d(u) > 1$
$d(v) > 1$

4. edge-coloring on $G \leftrightarrow$ vertex coloring of $L(G)$

Edge-coloring (colors)
- assigning labels to each edge in G

Proper: no two edges that share an endpoint have the same color

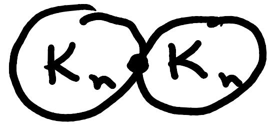
G L(G)

Edge-chromatic number $\chi'(G)$ is the minimum number of colors to properly edge-color G

Let's get boundin'

- $\chi'(G) \geq \Delta(G)$, as the largest degree vertex in G requires different colors for all incident edges

$d(u)=\Delta(G)$ $d(v)=\Delta(G)$



- $\chi'(G) \leq 2\Delta(G)-1$ via the worst case with greedy edge-coloring
- $\chi'(G) = \Delta(G)$ if G is bipartite

Note: k-regular graphs have a perfect match

↳ all bipartite graphs are a subgraph of a k-regular graph

↳ we can color our P.M. with the same color, remove and repeat k times to get a $k=\Delta(G)$-edge-coloring a $k = \Delta(G)$ - edge-coloring

---

We can actually tighten up our upper bound by quite a bit

Prove: $\chi'(G) = \Delta(G)$

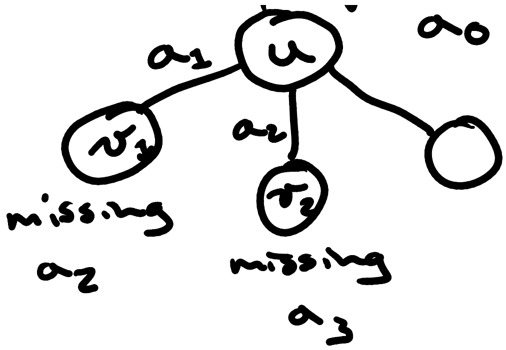
or $\Delta(G) + 1$

for simple graph G

PROOF BY ALGO.

- consider f as a $\Delta(G)+1$ edge-coloring of some subgraph $H \subseteq G$
  - $\rightarrow$ extend f to all of G
- consider $u \in V(G)$ and uncolored edge $(u, v_0) \in E(G)$
- In $N(u)$, there are some colors missing, a_0 is one such color

- consider u's neighbors
- label our neighbors such that $a_{i-1} \rightarrow$ color



such that $a_{i+1} \rightarrow$ color missing at v_i

- If color a_0 is not in v_0's neighborhood $\rightarrow$ color (u, v_0) with color a_0
- If color a_1 is not in v_0's neighborhood and a_1 is not in u's neighborhood
  $\rightarrow$ color (u, v_0) with a_1
- If a_2 is missing at v_1, there must exist (u, v_2) with color a_2 otherwise we can replace a_1 with a_2 and color (u, v_0) with a_1

Generally: if a_i is missing, we can use a_i on (u, v_{i-1}) and "shift" our colors down to eventually color (u, v_0) with a_1

$\rightarrow$ either a <u>missing color</u> repeats

→ either a <u>missing color</u> repeats
or this procedure is possible,
since we have at most $\Delta(G)+1$
colors

- v_ℓ is the first vertex with a
missing color on $a_1 \dots a_\ell$, we'll
call this color a_k

Note: also missing at v_{k-1} and it
appears on edge (v_k, u)

Note x2: a_0 also appears on v_ℓ
otherwise we could color (u, v_ℓ)
with a_0 and shift colors down

Consider P as a maximal a_0, a_k-
alternating path from v_ℓ

Case 1: P reaches v_k

→ shift colors down from v_{k-1}
and swap colors on P

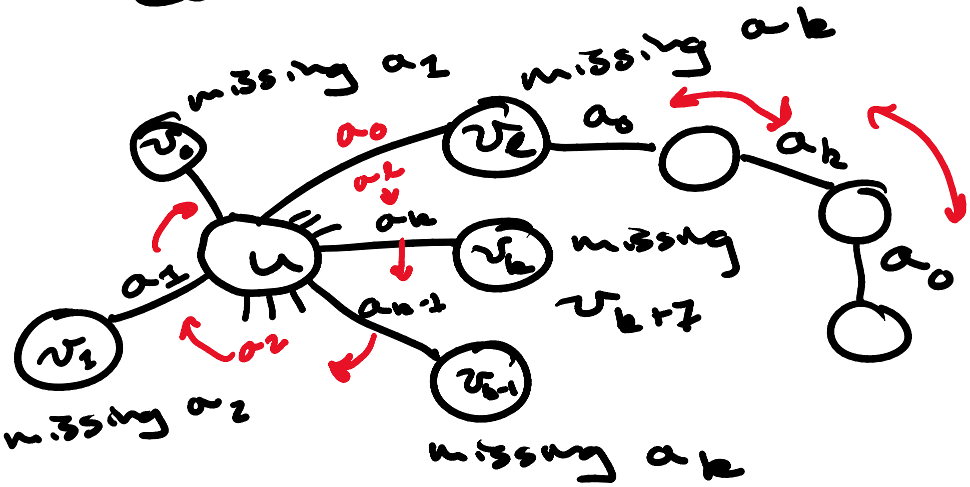
Case 2: P reaches v_{k-1}

Case 2: P reaches v_{k-1}

→ shift colors down from v_{k-1}, put a_0 on (u, v_{k-1}) and swap colors on P

Case 3: P reaches elsewhere

→ shift colors down from v_ℓ, put a_0 on (u, v_ℓ) and swap colors on P



→ all simple graphs have an edge chromatic number of $\underbrace{\Delta(G) \text{ or } \Delta(G)+1}$ ☑

As we've discussed,

if $\exists H$ s.t. $L(H) = G$

→ we can get a maximum independent set on G in polynomial time (normally exponential)

→ we can get an optimal coloring on G in quadratic time

$O(n)$ linear

P $O(n^k)$ polynomial

$\stackrel{?}{=}$

NP $O(2^n)$ exponential

Big Q: For what Gs does such an H exist

→ $L(H) = G$

For simple graph G, $\exists H$ s.t. $L(H) = G$ iff G decomposes into maximal cliques with each $v \in V(G)$ being in at most 2 being in at most 2

$\exists H \Rightarrow G$ decomposes

Note: every vertex in H becomes a clique in G

And: every edge in H is attached to exactly two vertices

edges in H $\to$ vertices in G ✓

G decomposes $\Rightarrow \exists H$

define: $S_1 S_2 \dots S_k$ as vertex sets of maximal cliques in a decomposition of G

To construct H:

$v_1 v_2 \dots v_e$ are vertices in only one of S_i

H gets one vertex for each in $\{S_1 \dots S_k, v_1 \dots v_e\} = V(H)$

H gets edges between each (v_i, S_j) and (S_x, S_y) where these vertex sets intersect

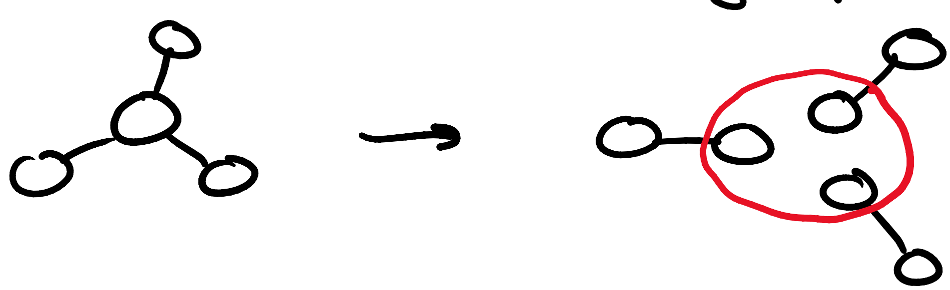
$\rightarrow$ each $v \in V(G)$ in at most two sets, with no two vertices in the same two sets

$\Rightarrow$ together, this implies the existence of H s.t.

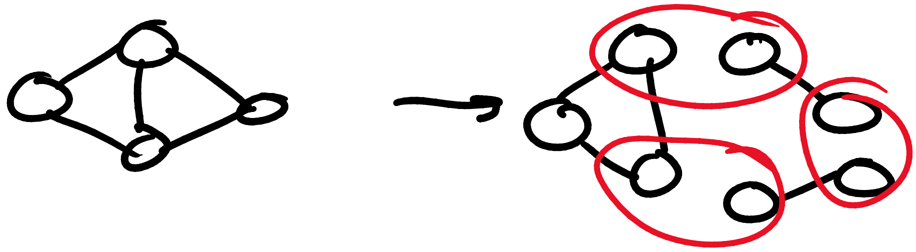
$L(H) = G$ $\square$

Is there an easier characterization of when $\exists H$ s.t. $L(H) = G$

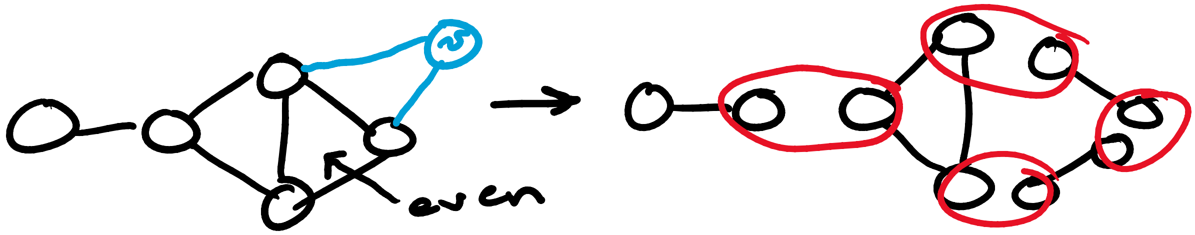
consider a claw graph



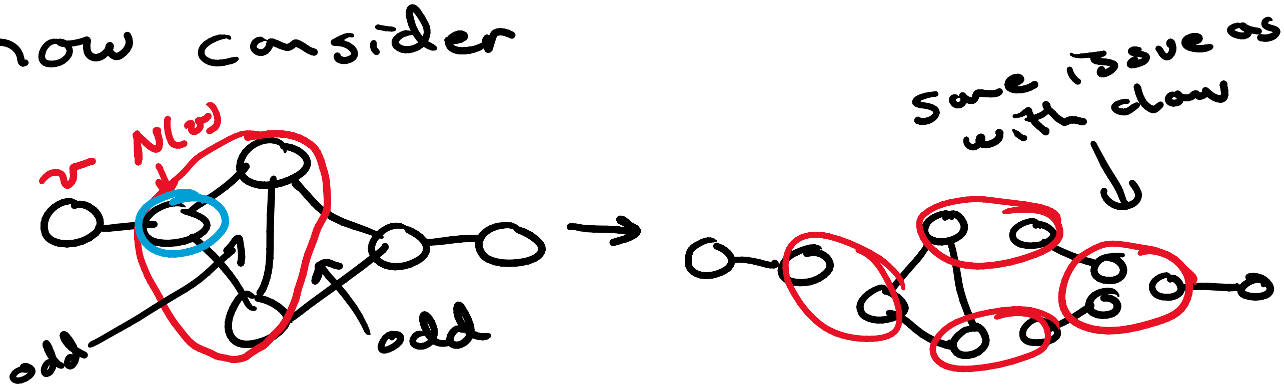
$\Rightarrow$ no G can have a claw graph as an <u>induced</u> subgraph consider a double triangle:



what about:



now consider



odd triangle T: $\exists v \in V(G)$

s.t. $|N(v) \cap V(T)| = \text{odd}$

even triangle T: $\exists v \in V(G)$

s.t. $|N(v) \cap V(T)| = \text{even}$

$\Rightarrow$ no G can have a double odd triangle as an induced subgraph if $\exists H$ s.t. $L(H) = G$

subgraph if $\exists H$ s.t. $L(H)=G$

Note: we've demonstrated necessity, but are they also sufficient?

$\exists H$ s.t. $G = L(H)$ iff

G has no claws or double odd triangles (O.O.T.S)

$\exists H \Rightarrow$ G has no claws or O.O.T.s

contrapositive

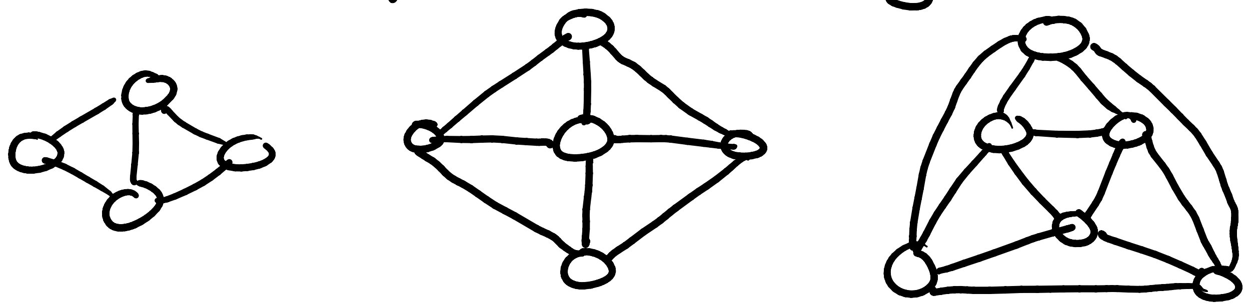
G has a claw or OOTs $\Rightarrow$ no such H

$\rightarrow$ we just showed this ✓

G has no claws or OOTs $\Rightarrow \exists H$

First: consider double even triangles

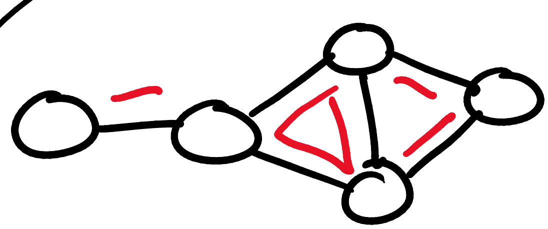
→ for simple graphs, only 3 exist



=> so we only need to consider graphs with double triangles that have one odd and one even triangle

- consider a maximal clique decomposition, with one special caveat:

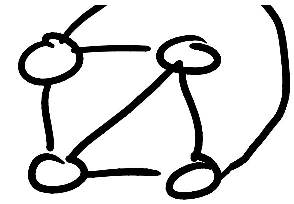
$S_1 \dots S_k$ are maximal cliques <u>except</u> for even triangles that aren't shared with an odd triangle



Note: for all K_n: $n>3$ they are comprised of odd triangles

→ we have a proper

→ we have a proper decomposition



Question: $\forall v \in V(G)$, is v in at most 2 subgraphs in our decomposition?