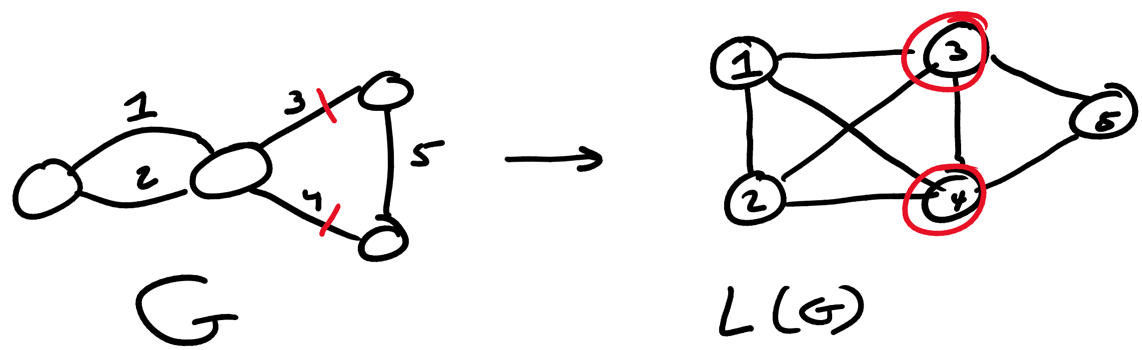
Line Graphs

Tuesday, April 20, 2021 2:22 PM

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

Edges of $G \to$ vertices of $L(G)$

Edges in $L(G)$ exist where edges in G share an endpoint



The equivalence of edges $\to$ vertices is relevant to several problems:

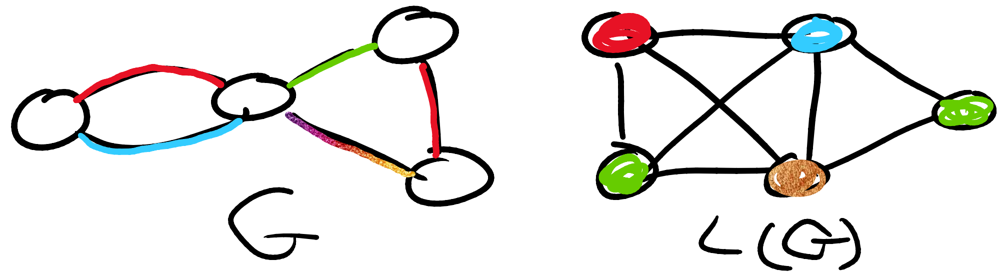
- Euler Tour on $G \to$ spanning cycle on $L(G)$
- * Matching on $G \to$ independent set on $L(G)$
- Cut edge on $G \to$ cut vertex on $L(G)$
  $e = (u, v)$ if $d(u) > 1$, $d(v) > 1$
- Edge-coloring on $G \to$ vertex coloring on $L(G)$

Edge-coloring

- assigning labels to each edge in G

- assigning labels to each edge in G (colors)

→ proper edge-coloring: no two edges that share an endpoint have the same color



- 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 requires different colors for all incident edges

$$(\Delta(G)-1)+(\Delta(G)-1)+1$$

- $\chi'(G) \leq 2\Delta(G)-1$, via our greedy edge-coloring algorithm greedy edge-coloring algorithm

- $\chi'(G) = \Delta(G)$ if G is bipartite

  Note: k-regulars graphs have a P.M.

  all bipartite graphs are subgraphs of a k-regular graph

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

Let's tighten this upper bound

Prove: $\chi'(G) \leq \Delta(G) + 1$

$\chi'(G) \geq \Delta(G)$ for simple graph G

- Consider f as $\Delta(G)+1$ coloring of some subgraph $H \subseteq G$

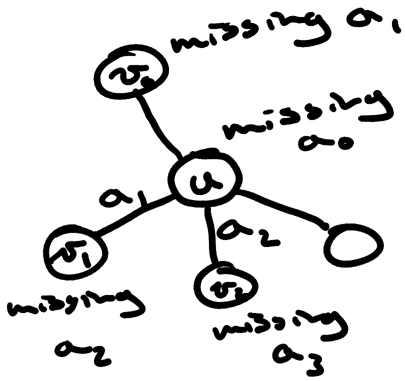
  → extend f to all of G

- consider $u \in V(G)$ and uncolored edge $(u, v_0) \in E(G)$

- In $N(u)$, there are some colors

  ... such color

- In $N(u)$, there
  missing, a_0 as one such color



- consider u's neighbor
- label our neighbors

★ such $a_{i+1} \to$ color missing at v_i

- If color a_0 is not in v_0's neighborhood
  $\to$ color (u, v_0) with color a_0

- If color a_1 is missing at v_0 and a_1 is not in $N(u)$
  $\to$ 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 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

$\to$ either a missing color repeats

→ either a missing color repeats

or this is possible, since we have at most $\Delta(G)+1$ colors

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

  Note: also missing at v_{k-1} and 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 down*

- P is 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

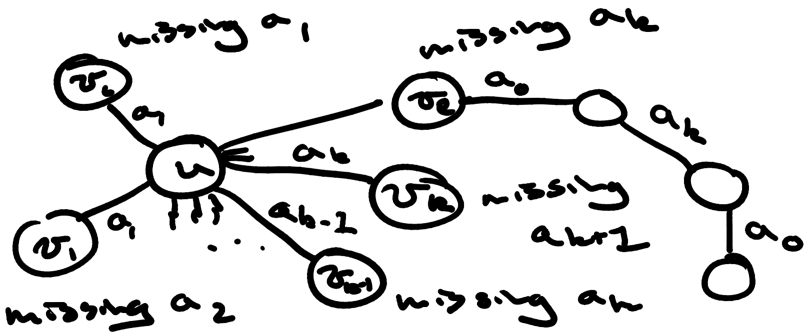
Case 2: P reaches v_{k-1}

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

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
edge-chromatic number of
either $\underbrace{\Delta(G) \text{ or } \Delta(G)+1}$

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 time)

(normally exponential time)

→ we can get a near-optimal coloring on G in quadratic time

$O(n)$ linear

P $O(n^k)$ polynomial

NP $\boxed{O(2^n) \text{ exponential}}$

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

$\exists H \Rightarrow G$ decomposes

Note: every vertex in H becomes a clique in G a clique in G

And: every edge in H is attached to two vertices

edges in $H \rightarrow$ vertices in G

G decomposes $\Rightarrow \exists H$

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

To construct $H \rightarrow v_1 v_2 \dots v_e$ are vertices only in one S_i

H then gets one vertex for each in $\{S_1 \dots S_k, v_1 \dots v_e\}$

H gets edges between each (v_i, S_j) and (S_a, S_b) where these intersect

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

→ this implies the existence of H

$\square$

QEWL

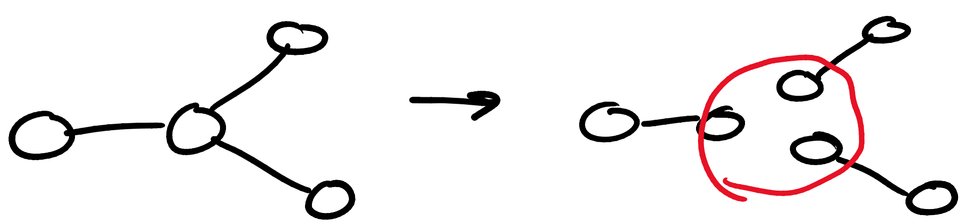
---

when does $\exists H$ s.t. $L(H) = G$?

→ G decomposes into cliques with each vertex in at most 2 cliques

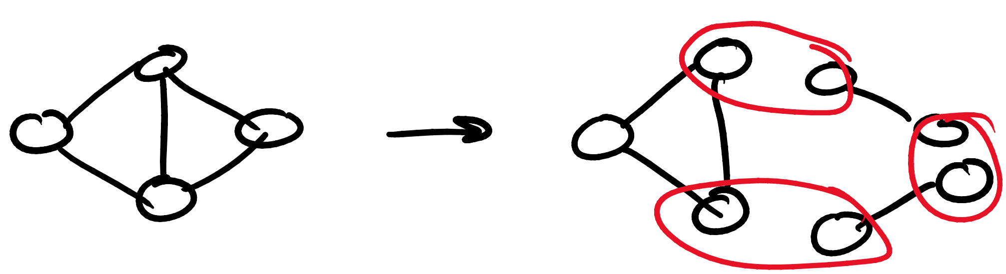
Is there an easier characterization?

consider a claw graph:

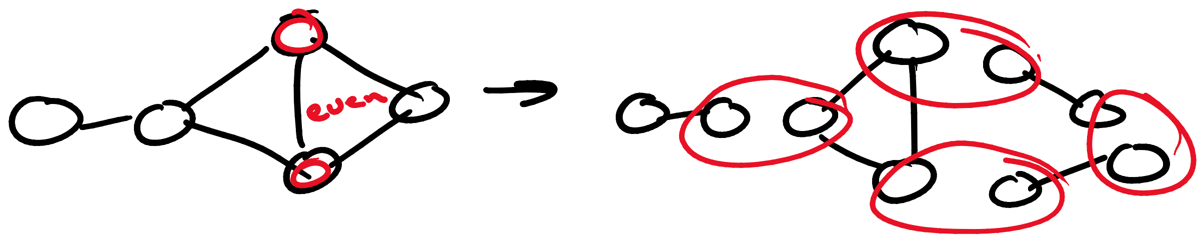


⟹ no G can have a claw graph as an <u>induced</u> subgraph

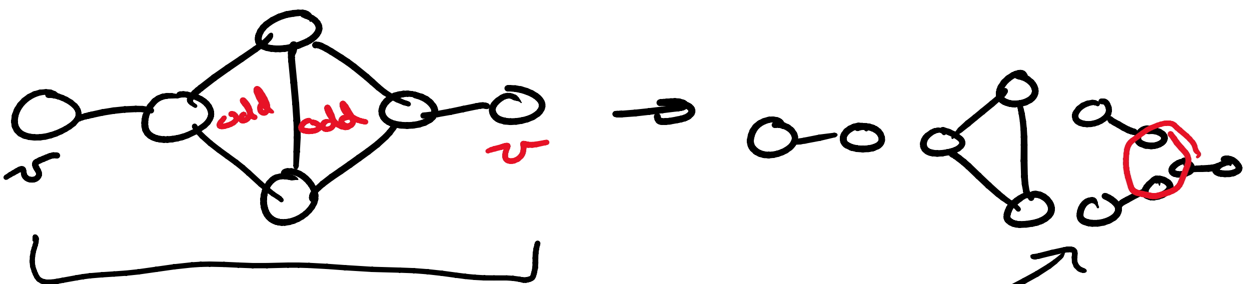
Consider a double triangle:



what about:



now consider:



Same issue as with the claw

odd triangle T: $\exists v \in V(G)$ s.t. $|N(v) \cap V(T)| = \text{odd}$

even triangle T: $\forall v \in V(G)$ $|N(v) \cap V(T)| = \text{even}$

$\Rightarrow$ no G can have a double odd triangle as an induced subgraph

subgraph

Note: these two conditions are necessary, but are they also sufficient?

$\exists H$ s.t. $G = L(H)$ iff G has no claws or double odd triangle

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

Contrapositive

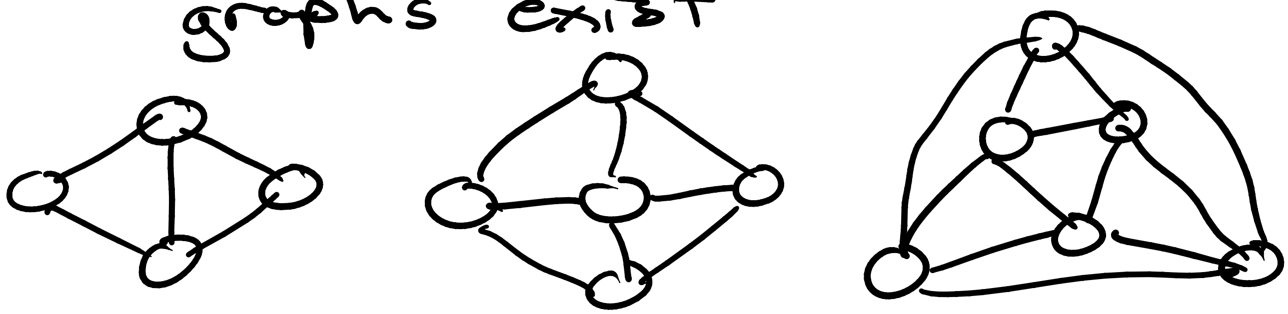
G has a claw or DOT $\Rightarrow$ no such H $\Rightarrow$ we already showed this ✓

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

First: consider double even triangles

First: consider double even triangles

→ for simple graphs, only 3 such graphs exist



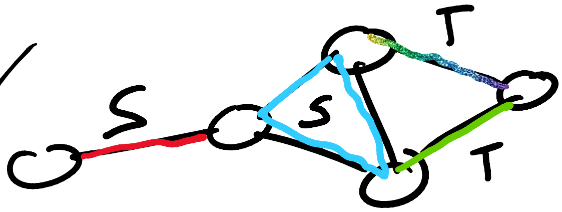
⇒ so we only need to consider graphs with double triangle that have one odd and one even

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

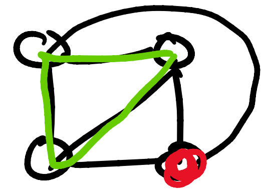
$S_1 \dots S_k$ are maximal cliques <u>except</u> for even triangles

$T_1 \dots T_\ell$ are edges in even triangles that aren't shared with an odd triangles

Note: for all K



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



we have a proper decomposition

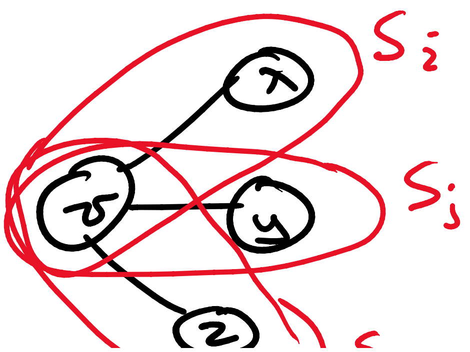
Question: $\forall v \in V(G)$, is v in at most 2 subgraphs in our decomposition?

- consider $v \in S_i, S_j, S_k$

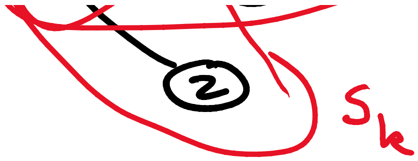
$x \in S_i$, $y \in S_j$, $z \in S_k$

$\{x, y, z\} \in N(v)$

Case 1: no edges $(x,y), (y,z), (x,z)$

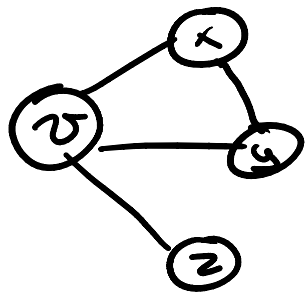


→ a claw, so some edge must



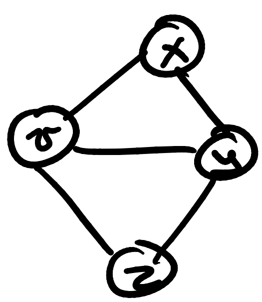
some edge must exist

Case 2: edge (x, y) exists



→ an odd triangle, but our decomposition specified an odd triangle is in one S_ℓ

Case 3: edges (x, y) and (y, z) exist



→ we have an two even triangles, regardless of choice of v

Case 4: edges (x, y), (y, z), (z, x)

→ we have K_4 which would be contained in some S_ℓ

From our prior proof → existence

From our prior proof → existence of decomposition, and each vertex can only be in at most 2 subgraphs

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

Qewl

↳ Characterization of G s.t. $G = L(H)$

G has no claws
G has no double odd triangles

↳ FORBIDDEN SUBGRAPHS

If G contains a forbidden subgraph
→ G can't decompose

$\Rightarrow$ there doesn't exist a H s.t. $G = L(H)$