$\chi(G,k)$ = # of ways to properly color G given k colors

obviously
$\chi(G,k)=0$ if $k<\chi(G)$

consider clique K_n and some k

How do we determine $\chi(K_n,k)$?

→ we can apply basic logic to reason our way through it

First, color any $v \in V(K_n)$ with one of the possible k colors

Next, color some other $u \in V(K_n)$ $u \neq v$
→ we have $(k-1)$ valid choices

Next, color some other $w \in V(K_n)$ $w \neq u \neq v$

→ $(k-2)$ choices

$w \neq u \neq v$

$\rightarrow (k-2)$ choices

$\ldots (k-3)$

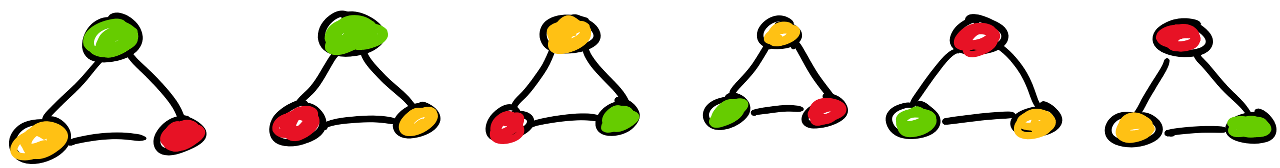
$\ldots (k-4)$

$\bullet \bullet \bullet$

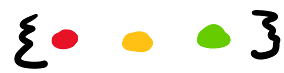
color final vertex with

$(k-n+1)$ possible colors

$$\chi(K_n, k) = k(k-1)(k-2)\ldots(k-n+1)$$



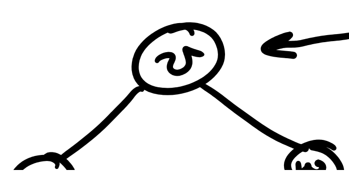
consider K_3 and $k=3$



Let's talk trees

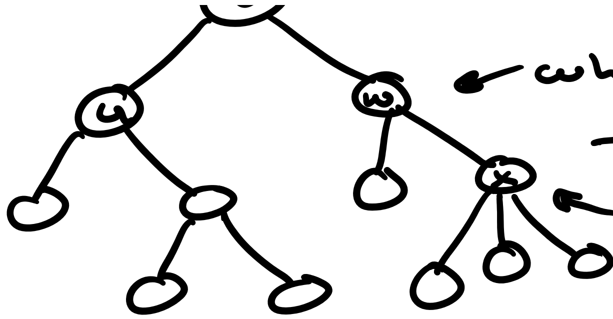
- Consider tree T and some root $v \in V(T)$ and a BFS from v

breadth-first search



$\leftarrow$ can use any of k colors

$\leftarrow$ what about vertex w?



← what about vertex w?

→ any of $(k-1)$ choices

← what about x?

→ any of $(k-1)$ choices

Consider coloring T level by level

→ we're only restricted for the color of some child by the color of its parent

$\Rightarrow$ all children have $(k-1)$ choices

$$\chi(T, k) = k(k-1)^{n-1}$$

Generally, we refer to $\chi(G, k)$ as the

Chromatic Polynomial

General form:

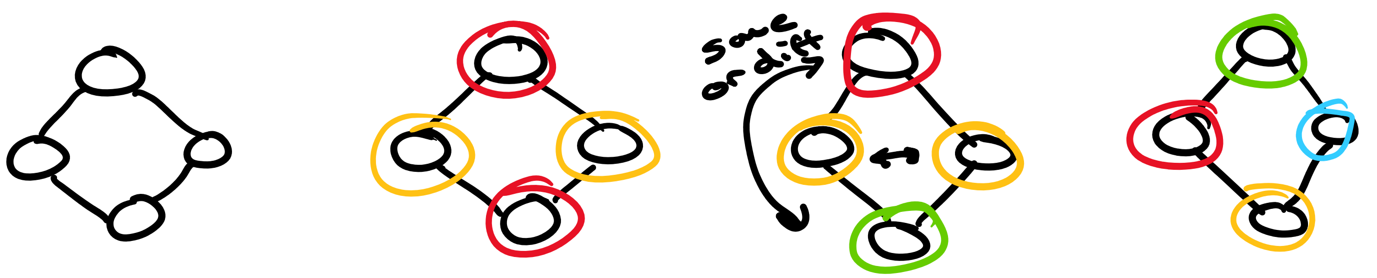
$$\chi(G, k) = \sum_{r=1}^{n} p_r(G)\, k_r$$

$P_r(G)$ = # of ways to partition $V(G)$ into r independent sets

k_r = # of ways to color r independent set with k colors

$$k_r = k(k-1)(k-2)\ldots(k-r+1)$$

Consider C_4 and its chromatic polynomial $\chi(C_4, k)$



$P_1(C_4)=0$ $\quad P_2(C_4)=1$ $\quad P_3(C_4)=2$ $\quad P_4(C_4)=1$

$$\chi(G,k) = \sum_{r=1}^{n} P_r(G)\, k_r$$

$$= \underbrace{0}_{r=1} + \underbrace{1(k)(k-1)}_{r=2}$$

$$+ \underbrace{2(k)(k-1)(k-2)}_{r=3}$$

$$+\underbrace{1(k)(k-1)(k-2)(k-3)}_{r=4}$$

($r=3$)

$$\chi(C_4,k) = k(k-1) + 2k(k-1)(k-2) + k(k-1)(k-2)(k-3)$$

Note: we can use the chromatic polynomial to get $\chi(G)$

$$\chi(C_4, k=0) = 0$$

$$\chi(C_4, k=1) = 1(0) + 2(1)(0)(-1) + (1)(0)(-1)(-2) = 0 \checkmark$$

$$\chi(C_4, k=2) = 2(1) + 2(2)(1)(0) + 2(1)(0)(-1) = \underbrace{2}_{} \checkmark$$

expected, as we know $\chi(C_4) = 2$

Q: Can we derive $\chi(G,k)$ in a simpler way?

in a simpler way?

A: not really

$\rightarrow P_r(G)$ is "tough" to compute

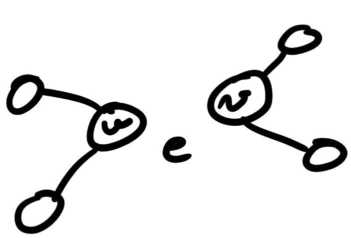
But $\rightarrow$ we can also determine $\chi(G,k)$ in a different way that doesn't need $P_r(G)$

Fundamental Reduction Theorem

$$\chi(G,k) = \chi(G-e,k) - \chi(G\cdot e,k)$$

$e = (u,v) \in E(G)$

$\chi(G-e,k)$ = # of ways to color G

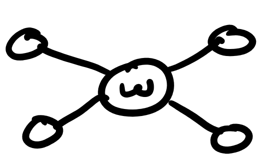


s.t. $c(u) = c(v)$

$c(u) \neq c(v)$

(simple graphs)

$\chi(G\cdot e,k)$ = # of ways to color G



s.t. $c(u) = c(v)$

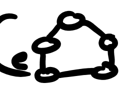
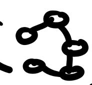
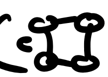
$\chi(G,k) = \chi(G-e,k) - \chi(G\cdot e,k)$

$$\chi(G,k) = \underbrace{\chi(G-e,k) - \chi(G\cdot e,k)}_{\#\text{ of ways to color } G \text{ s.t. } c(u)\neq c(v)}$$

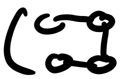
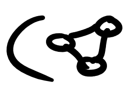
Recall: $\chi(K_n, k) = k(k-1)\dots(k-n+1)$

$\chi(T, k) = k(k-1)^{n-1}$

Consider C_5 and the above

$\chi($  $, k) = \chi($  $, k) - \chi($  $, k)$

tree

$= k(k-1)^4 - [\chi($  $, k) - \chi($  $, k)]$

tree   clique

$= k(k-1)^4 - k(k-1)^3 + k(k-1)(k-2)$

Let's determine $\chi(C_5)$

$\chi(C_5, k=0) = 0$

$\chi(C_5, k=1) = 0$

$$\chi(C_5, k=2) = \underbrace{2(1)^4 - 2(1)^3}_{=0} + 2(1)(0)$$

$= 0$

$=0$

$= 0$

$$\chi(C_5, k=3) = 3(2)^4 - 3(2^3) + 3(2)(1)$$

$$= 48 - 24 + 6$$

$$= 30 \checkmark$$

Let's check ✓ it out

→ C_5 has 5 vertices

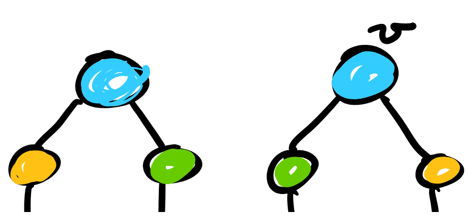
→ we also know C_5 is color-critical

↳ so each vertex can be colored uniquely (sp?)

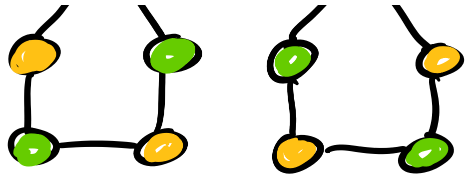
so algorithmically

→ each of 5 vertices can get one of 3 possible colors

$\Rightarrow 15$ choices



$\Rightarrow$ 2 choices for $N(v)$



$\Longrightarrow$ 2 choices for $N(v)$

(rest of colors only depend on v and $N(v)$)

Together

$\longrightarrow$ (15 choices)(2 choices)

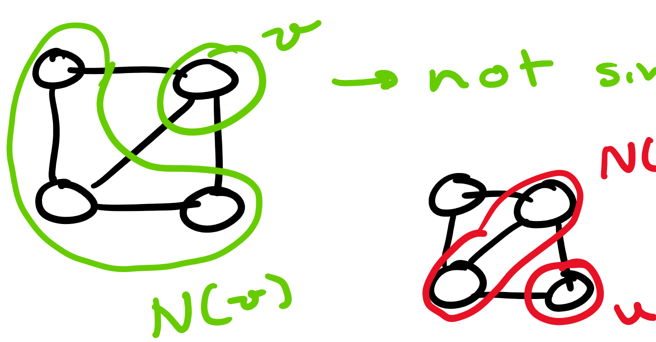
$= 30$ ways to color C_5 with $k=3$ colors ✓

Simplicial vertices

A <u>simplicial vertex</u> is a vertex v where $N(v)$ is a clique

$\downarrow$

includes K_0, K_1, K_2

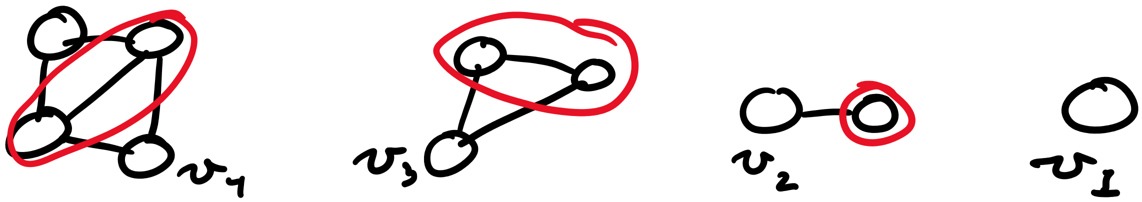


$\longrightarrow$ not simplicial

<u>simplicial elimination ordering</u> (SEO)

an ordering $\{v_n\, v_{n-1} \dots v_1\}$ of all $v \in V(G)$ for deletion, such that

$v \in V(G)$ for deletion, such that each v_i is simplicial on induced graph $G[\{v_i\, v_{i-1} \dots v_1\}]$



$$\{v_4\ v_3\ v_2\ v_1\}$$

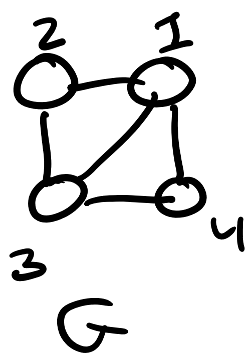
Why is this relevant?

→ we can use SEO to get $\chi(G,k)$

How? add $v_1 v_2 v_3 \dots v_i$ to $G_i : G[\{v_1 \dots v_i\}]$

$$\chi(G,k) = \prod_{i=1}^{n} (k - d'(v_i))$$

degree of v_i in G_i



$d'(v_1) = 0$

$d'(v_2) = 1$

$d'(v_3) = 2$

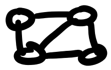
$d'(v_4) = 2$

$\Rightarrow \chi(G,k) = k(k-1)(k-2)^2$

$\hookrightarrow \chi(G,k) = k(k-1)(k-2)^2$

Why does this work?

$\rightarrow$ same logic as before, we can guarantee some $k - d'(v_i)$ color choices as $N(v_i)$ is a clique and therefore will have $|N(v_i)|$ unique colors

And to get $\chi($  $)$:

$\chi(G,0) = 0$

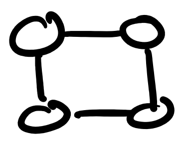
$\chi(G,1) = 0$

$\chi(G,2) = 0$

$\chi(G,3) = 6$

Note: $\omega(G) = 3$

$\uparrow$ recall: clique number

Also Note:  has no SEO

(?): what graphs have a SEO?

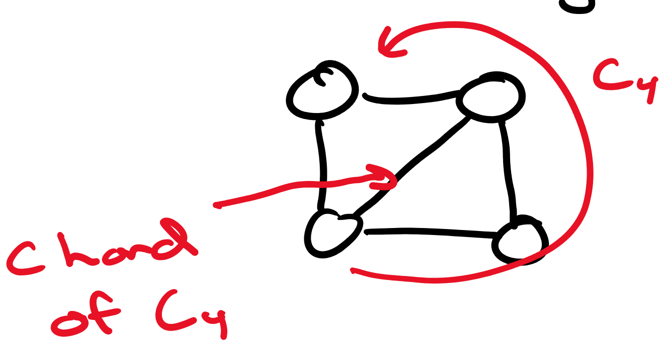
Q: What graphs have a SEO?

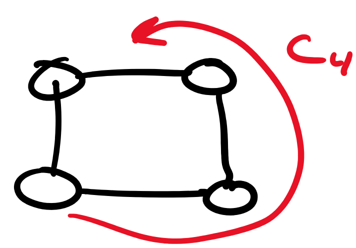
A: chordal graphs

chordal graph: a simple graph that has no chordless cycles

Chordless cycle: a cycle of at least 4 edges with no chords

Chord: an edge with both endpoints on same cycle but is not part of that cycle





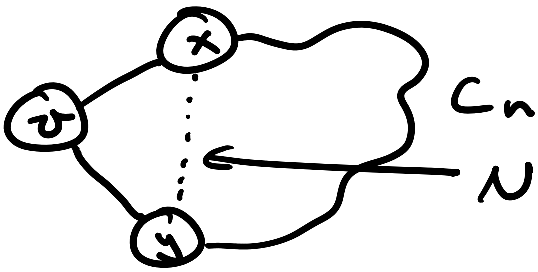
chordless cycle

G has an SEO $\Leftrightarrow$ G is chordal

G has an SEO $\iff$ G is chordal

($\Rightarrow$) Consider some $C_{n \geq 4} \subseteq G$

Consider the first $v \in V(C_n)$ eliminated in the SEO



Note: if $(x,y) \notin E(G)$ v would not be simplicial

($\Leftarrow$)

Show: every chordal graph has a simplicial vertex

Note: deleting a vertex can't introduce a chordless cycle

Strong induction on $|V(G)|$

Basis $P(1) \rightarrow$ single vertex is simplicial

$P(n>1)$: We have G, $|V(G)| = n$

$\rightarrow$ consider $x \in V(G)$ and $G-x$

Case 1: $N(x) = \{V(G) - x\}$

$G - x$ is chordal

I.H. on $G - x$

$\rightarrow$ any simplicial vertex on $G - x$
is simplicial on G

$SEO(G) = \{x\} + SEO(G - x)$

Case 2: $N(x) \neq \{V(G) - x\}$

define $T = \{$vertices of maximum distance from $x\}$

define H = subgraph induced on $T \rightarrow G[T]$

define S = vertices in $G - T$ with neighbors in $V(H) \cup T$

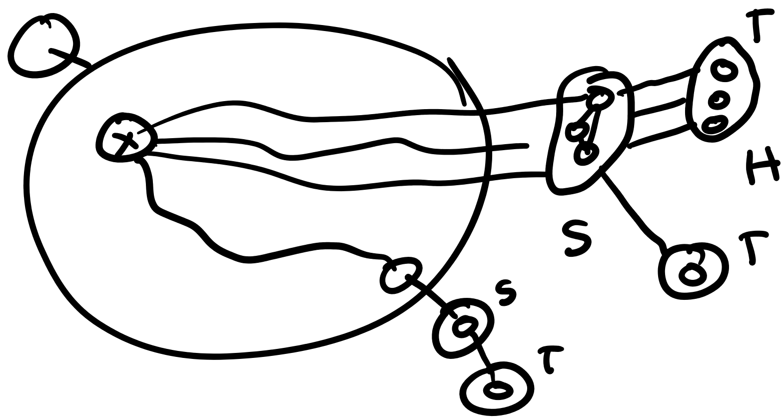
define Q = component of $G - S$ that contains x

Note: S must be cliques (induced subgraph)

$\rightarrow$ has neighbors in Q and H for all $w \in S$

$\rightarrow$ any cycle from $H \rightarrow Q \rightarrow H$

→ any cycle from $H \to Q \to H$ passing through $u,v \in S$ must have (u,v) as a chord



define $G' = G[S \cup V(H)]$

→ I.H. on G'

Consider some $u \in S$

→ $\exists$ simplicial vertex in $H \to z$

↳ also simplicial in G

=> so we can construct a SEO using this vertex □

(apply same logic to $G - z$)

Recall G is perfect if $\chi(H) = \omega(H)$ $\forall H \subseteq G$

Chordal graphs => perfect

Exercise 4 reader