Chap 6 Graph Coloring

Yih-Lang Li (李毅郎)
Computer Science Department
National Yang-Ming Chiao-Tung University, Taiwan

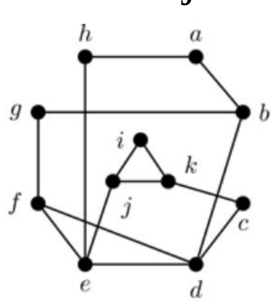
The sources of most figure images are from the textbook

Outline

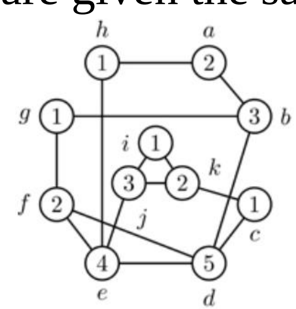
- Four Color Theorem
- Vertex Coloring
- Edge Coloring
- Coloring Variations

6.1 Four Color Theorem

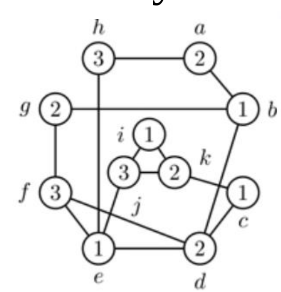
- Consider the following scenario as compared to previous matching problem
  - ✓ Five student groups are meeting on Saturday, with varying time requirements. The staff at the Campus Center need to determine how to place the groups into rooms while using the fewest rooms possible.
  - ✓ Still a matching problem.
  - ✓ How about minimize the number of used rooms? (resource constraint problem)
- In 1852 Augustus De Morgan sent a letter to his colleague Sir William Hamilton regarding a puzzle presented by one of his students, Frederick Gutherie.
  - ✓ Any map split into contiguous regions can be colored using at most four colors so that no two bordering regions are given the same color – Four Color Conjecture.
- **Definition 6.1** A proper k-coloring of a graph G is an assignment of colors to the vertices of G so that no two adjacent vertices are given the same color and exactly k colors are used.



G_1



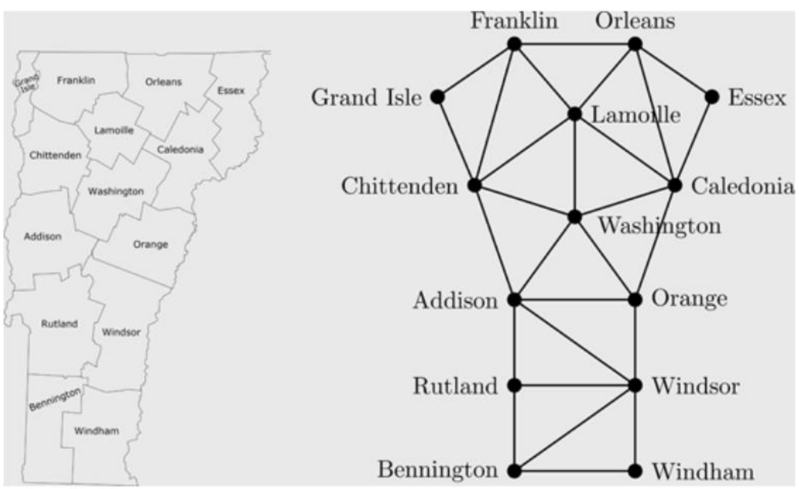
proper 5-coloring

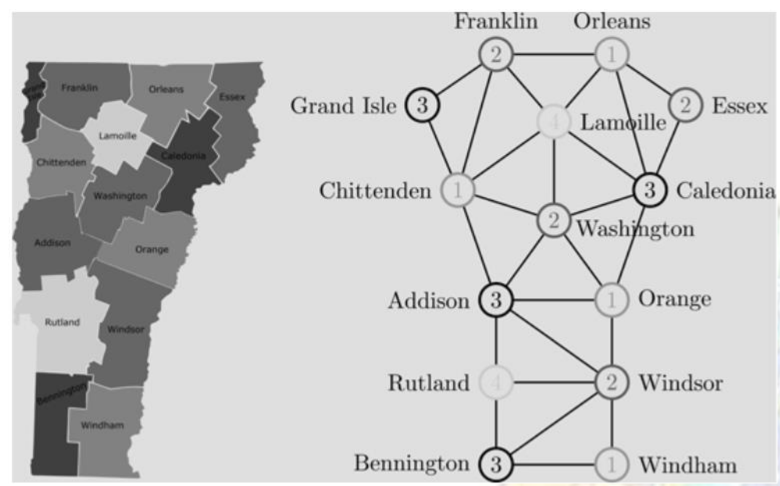


proper 3-coloring

Four Color Theorem

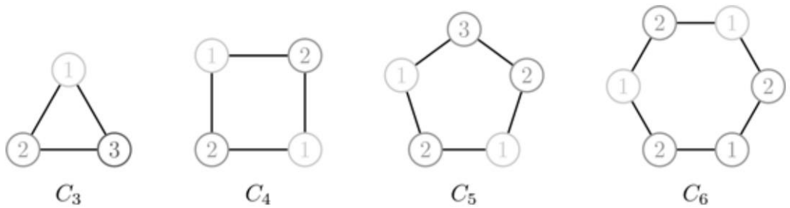
- **Definition 6.2** Given a proper k-coloring of G, the color classes are sets $S_1,\ldots S_k$ where S_i consists of all vertices of color i.
  - ✓ The first coloring of G_1 has color classes $S_1=\{c, g, h, i\}$, $S_2=\{a, f, k\}$, $S_3=\{b, j\}$, $S_4=\{e\}$, $S_5=\{d\}$ and the second coloring has color classes $S_1=\{b, c, e, i\}$, $S_2=\{a, d, g, k\}$, $S_3=\{f, h, j\}$.
- **Definition 6.3** The independence number of a graph G is $\alpha(G)=n$ if there exists a set of n vertices with no edges between them but every set of n+1 vertices contains at least one edge.
- **Example 6.1** Find a coloring of the map of the counties of Vermont and explain why three colors will not suffice.



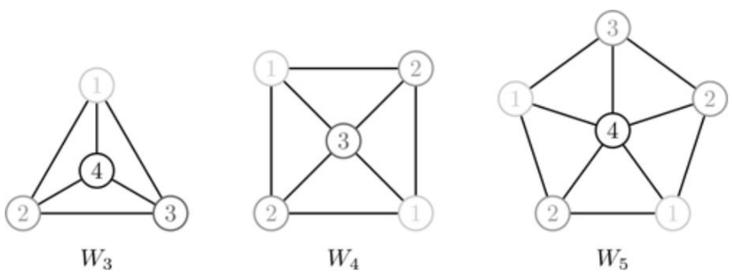


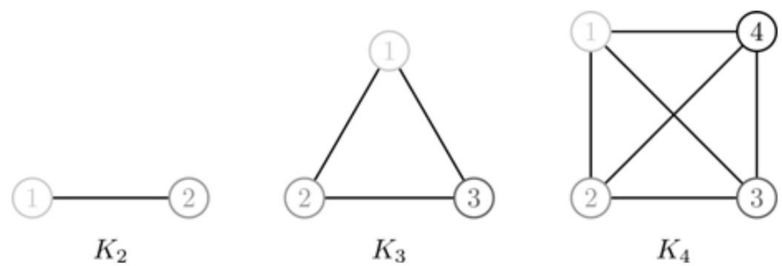
6.2 Vertex Coloring

- **Definition 6.4** The chromatic number $\chi(G)$ of a graph is the smallest value k for which G has a proper k-coloring.
- In order to determine the chromatic number of a graph, we often need to complete the following two steps:
  - ✓ Find a vertex coloring of G using k colors.
  - ✓ Show why fewer colors will not suffice.



- **Definition 6.5** A wheel W_n is a graph in which n vertices form a cycle around a central vertex that is adjacent to each of the vertices in the cycle.



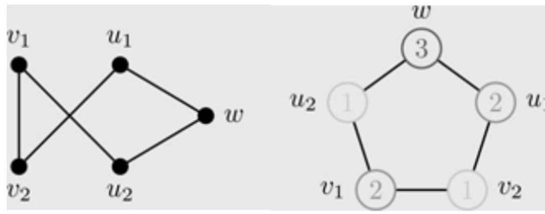


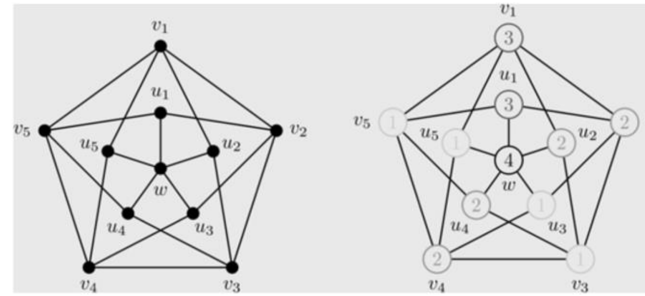
- **Definition 6.6** A clique in a graph is a subgraph that is itself a complete graph. The clique size of a graph G, denoted $\omega(G)$, is the largest value of n for which G contains K_n as a subgraph.

Vertex Coloring

- Special Classes of Graphs with known $\chi(G)$
  - ✓ $\chi(C_n)$=2 if n is even ($n \geq 2$)
  - ✓ $\chi(C_n)$=3 if n is odd ($n \geq 3$)
  - ✓ $\chi(K_n)=n$
  - ✓ $\chi(W_n)$=4 if n is odd ($n \geq 3$)
- A graph with small clique size must have small chromatic number?
  - ✓ Jan Mycielski showed that there exist graphs with an arbitrarily large chromatic number yet have a clique size of 2.
  - ✓ Graphs with $\omega(G)=2$ are referred as *triangle-free*.
- **Example 6.2** Mycielski's Construction is a well-known procedure in graph theory that produces *triangle-free* graphs with increasing chromatic numbers.
  - ✓ begin with a triangle-free graph G where $V(G)=\{v_1, v_2, \ldots, v_n\}$
  - ✓ add new vertices $U=\{u_1, u_2, \ldots, u_n\}$ so that $N(u_i)=N(v_i)$ for every i; that is, add an edge from u_i to v_j whenever v_i is adjacent to v_j.
  - ✓ add a new vertex w so that $N(w)=U$; that is, add an edge from w to every vertex in U.

Vertex Coloring

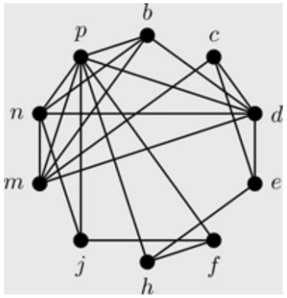


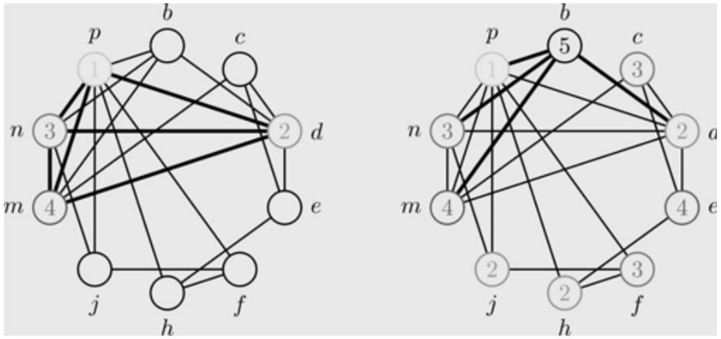


- **Theorem 6.7** (*Brooks Theorem*) Let G be a connected graph and Δ denote the maximum degree among all vertices in G. Then $\chi(G) \leq \Delta$ as long as G is not a complete graph or an odd cycle. If G is a complete graph or an odd cycle then $\chi(G)=\Delta+1$.
  - ✓ $\omega(G) \leq \chi(G) \leq \Delta(G)+1$
- Coloring strategies
  - ✓ large degree vertices are more likely to increase the value for the chromatic number of a graph and thus should be assigned a color earlier rather than later in the process.
  - ✓ In addition, it is better to look for locations in which colors are forced rather than chosen.
- **Example 6.3** Every year on Christmas Eve, the Petrie family compete in a friendly game of Trivial Pursuit. Unfortunately, due to longstanding disagreements and the outcome of previous years' games, some family members are not allowed on the same team.

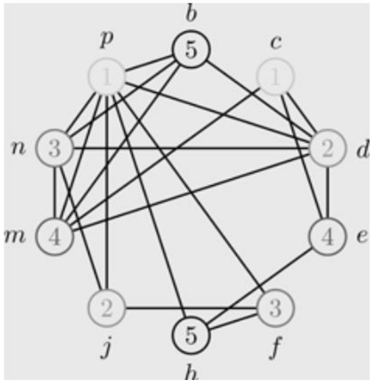
Coloring Strategies

	Betty	Carl	Dan	Edith	Frank	Henry	Judy	Marie	Nell	Pete
Betty	·	·	N	·	·	·	·	N	N	N
Carl	·	·	N	N	·	·	·	N	·	·
Dan	N	N	·	N	·	·	·	N	N	N
Edith	·	N	N	·	·	·	·	N	·	·
Frank	·	·	·	·	·	N	N	·	·	N
Henry	·	·	·	·	N	·	·	·	·	N
Judy	·	·	·	·	·	N	·	·	N	N
Marie	N	N	N	N	·	·	·	·	N	N
Nell	N	·	N	·	·	·	N	N	·	N
Pete	N	·	N	·	N	N	N	N	N	·





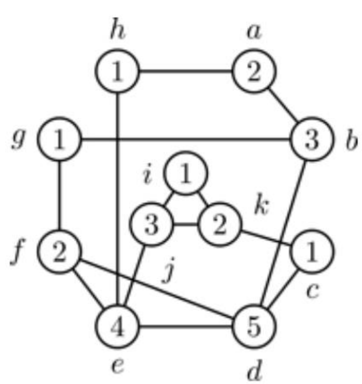
Team	Members		
1	Pete		
2	Dan	Henry	Judy
3	Carl	Frank	Nell
4	Edith	Marie	
5	Betty		

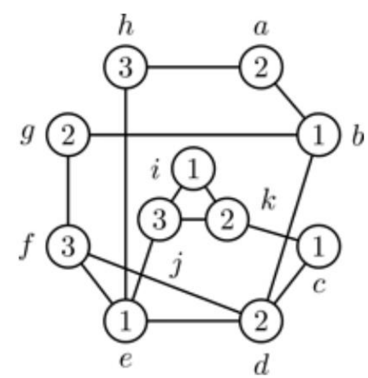


- Is the above coloring fair?
- **Definition 6.8** An equitable coloring is a minimal proper coloring of G so that the number of vertices of each color differs by at most one.
- **Example 6.4** Find an equitable coloring for the graph from Example 6.3.

Coloring Strategies

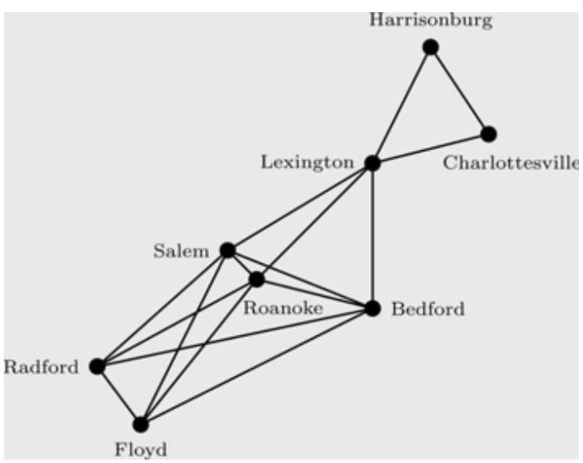
- A common coloring strategies
  - begin by finding all vertices that can be given color 1, and once that is done find all the vertices that can be given color 2, and so on.
  - The problem with this strategy is that you may choose to give vertex *x* color 1 which can necessitate the addition of a new color for vertex *y* when if *x* was given color 2, then *y* could be colored using one of the previously used colors.
    - with vertices *c*, *g*, *h*, *i* given color 1, followed by *a*, *f*, *k* given color 2
  - A better strategy is to focus on locations that *force* specific colors to be used rather than *choose* which color to use

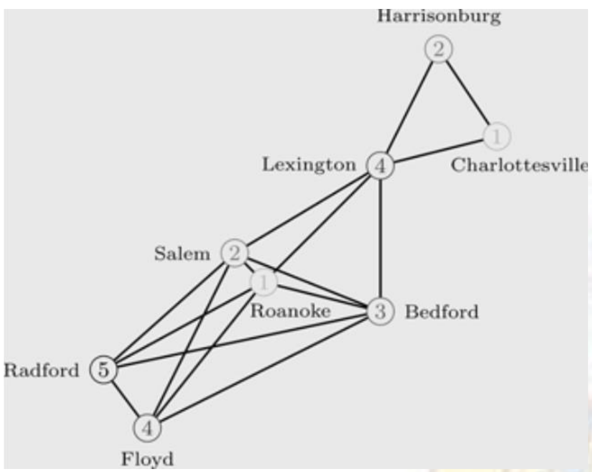




Coloring Strategies

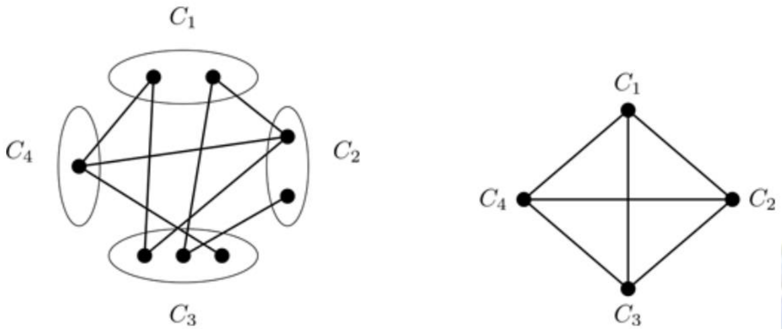
- *Basic Coloring Strategies*
  - ✓ Begin with vertices of high degree.
  - ✓ Look for locations where colors are forced (cliques, wheels, odd cycles) rather than chosen.
  - ✓ When these strategies have been exhausted, color the remaining vertices while trying to avoid using any additional colors.
- **Example 6.5** Due to the nature of radio signals, two stations can use the same frequency if they are at least 70 miles apart. An edge in the graph below indicates two cities that are at most 70 miles apart, necessitating different radio stations.





General Results

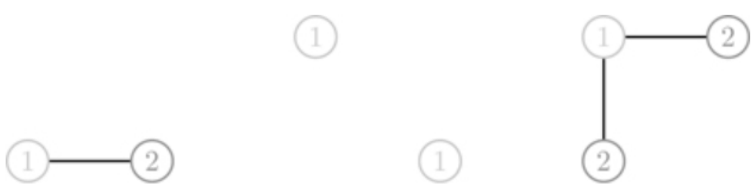
- **Proposition 6.9** Let G be a graph with m edges. Then $\chi(G) \leq \frac{1}{2} + \sqrt{2m + \frac{1}{4}}$
  - ✓ Assume $\chi(G)=k$. First note that there must be at least one edge between color classes.
  - ✓ Now, if we viewed each color class as a vertex and represented any edge between color classes as a singular edge in this new graph, we would obtain the complete graph K_k.
  - ✓ Thus G must have at least as many edges as the complete graph K_k, that is $m \geq \frac{k(k-1)}{2}$.
  - ✓ We can rewrite this as $2m \geq k(k-1) = k^2 - k$. Completing the square gives us $2m + \frac{1}{4} \geq k^2 - k + \frac{1}{4} = \left(k - \frac{1}{2}\right)^2$. Thus $k \leq \frac{1}{2} + \sqrt{2m + \frac{1}{4}}$.



- **Proposition 6.10** Let G be a graph and $l(G)$ be the length of the longest path in G. Then $\chi(G) \leq 1 + l(G)$.

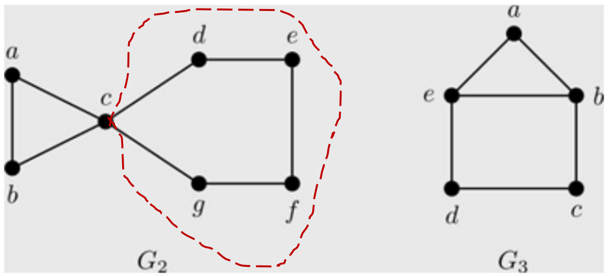
General Results

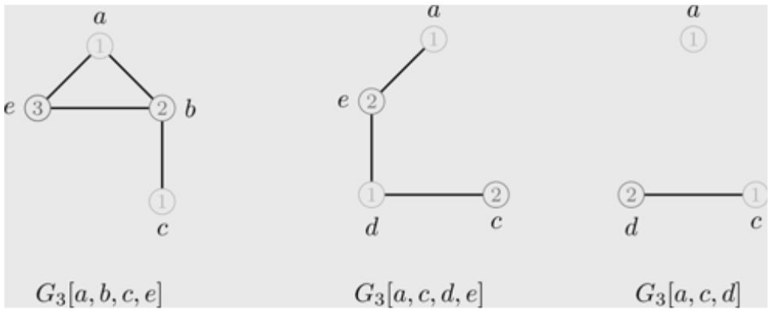
- **Definition 6.11** Given a graph $G=(V, E)$, an *induced subgraph* is a subgraph $G[V']$ where $V'\subseteq V$ and every available edge from G between the vertices in V' is included.
- **Proposition 6.12** Let G be a graph and $\delta(G)$ denote the minimum degree of G. Then $\chi(G)\leq 1+\max_H \delta(H)$ for any induced subgraph H.
- **Definition 6.13** A graph G is *perfect* if and only if $\chi(H)=\omega(H)$ for all induced subgraphs H.
  - ✓ even cycles are perfect but not odd cycles of length greater than 3.
  - ✓ First note that $\omega(C_n)=$__, for all values of $n\geq 4$.
  - ✓ any odd cycle of length at least __ cannot be perfect.
  - ✓ for the even cycles, a proper induced subgraph will either have an edge or a set of independent vertices $\rightarrow$ the induced subgraph will satisfy $\chi(H)=\omega(H)$.
  - ✓ Some sample induced subgraphs are shown for C_4 below.



General Results

- **Example 6.6** Determine if either of the two graphs below are perfect.
  - both graphs above satisfy $\chi(G)=\omega(G)$.
  - G_2 is not perfect. G_3 is perfect.

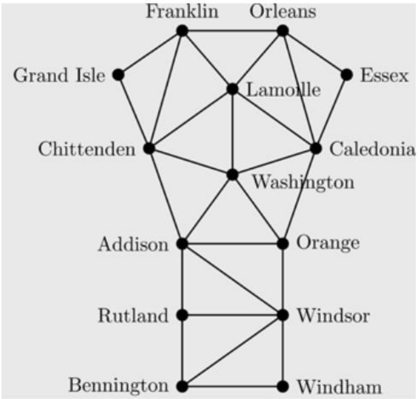


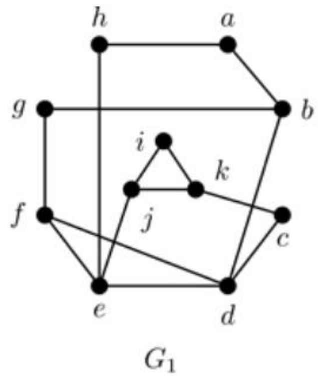


- **Theorem 6.14** A graph G is perfect if and only if $\bar{G}$ is perfect.
- **Theorem 6.15** A graph G is perfect if and only if no induced subgraph of G or $\bar{G}$ is an odd cycle of length at least 5.
- The following classes of graphs are known to be perfect:
  - Trees
  - Bipartite graphs
  - Chordal graphs
  - Interval graphs

General Results

- **Definition 6.16** A graph G is *chordal* if any cycle of length four or larger has an edge (called a *chord*) between two nonconsecutive vertices of the cycle.

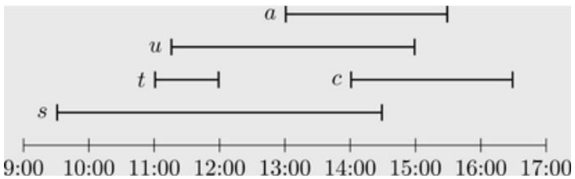


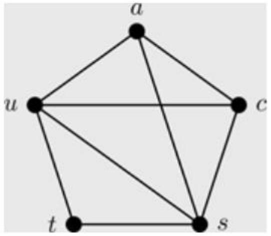


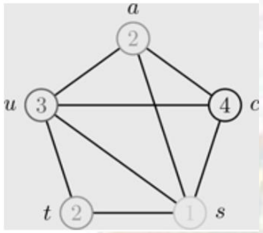
Student Group	Meeting Time
Agora	13:00–15:30
Counterpoint	14:00–16:30
Spectrum	9:30–14:30
Tupelos	11:00–12:00
Upstage	11:15–15:00

- **Definition 6.17** A graph G is an interval graph if every vertex can be represented as a finite interval and two vertices are adjacent whenever the corresponding intervals overlap; that is, for every vertex x there exists an interval I_x and xy is an edge in G if $I_x \cap I_y \neq \emptyset$.

- **Example 6.7** Five student groups are meeting on Saturday, with varying time requirements. The staff at the Campus Center need to determine how to place the groups into rooms while using the fewest rooms possible.

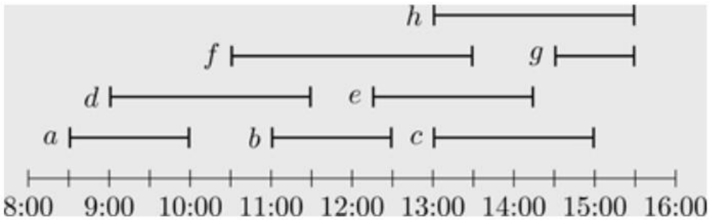


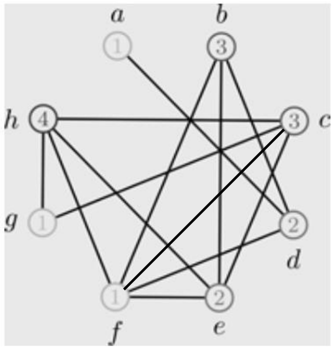




General Results

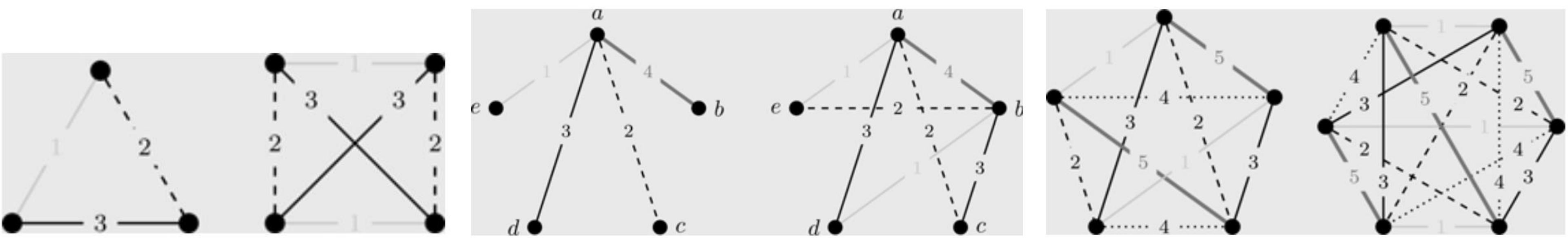
- **Example 6.8** Eight meetings must occur during a conference this upcoming weekend, as noted below. Determine the minimum number of rooms that must be reserved.





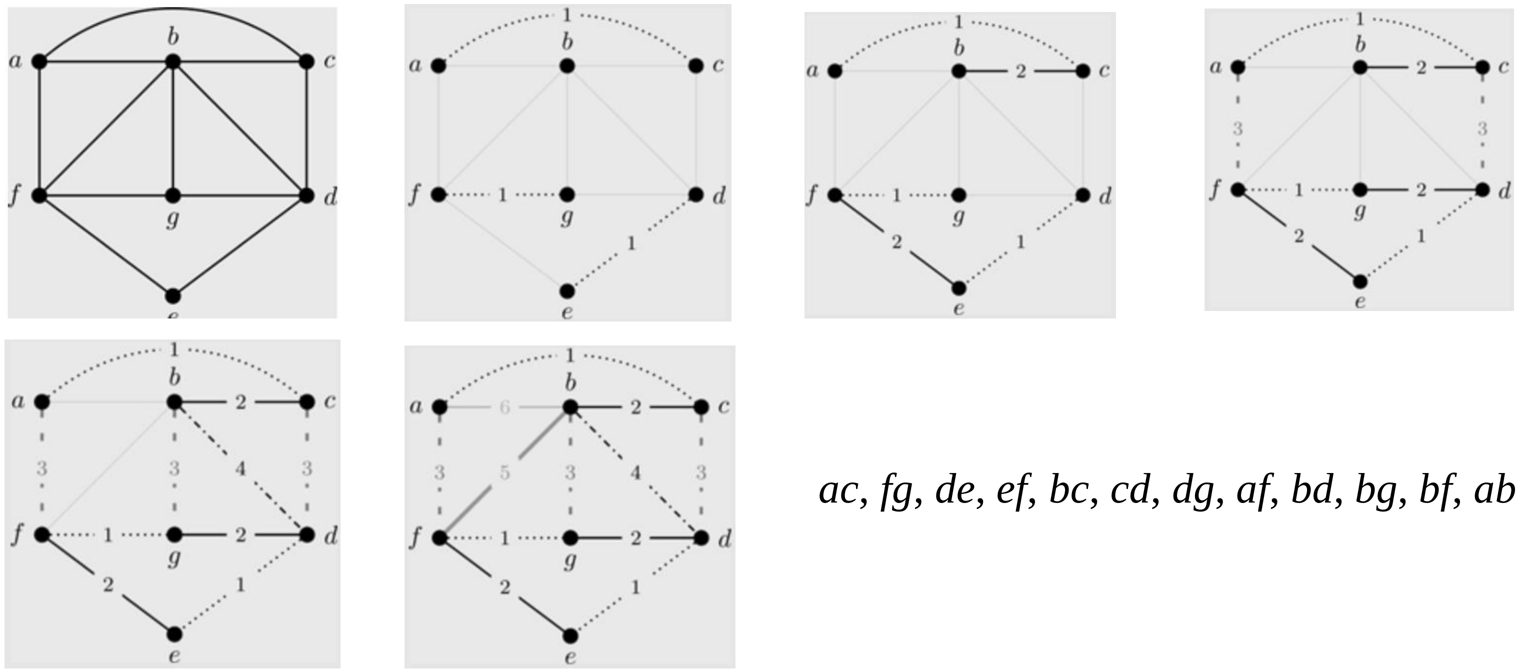
Edge Coloring

- **Definition 6.18** Given a graph $G=(V, E)$ an *edge-coloring* is an assignment of colors to the edges of G so that if two edges share an endpoint, then they are given different colors. The minimum number of colors needed over all possible edge-colorings is called the *chromatic index* and denoted $\chi'(G)$.
- **Example 6.9** Recall that the chromatic number for any complete graph is equal to the number of vertices. Find the chromatic index for K_n for all n up to 6.
  - ✓ $\chi'(K_1)=0$ and $\chi'(K_2)=1$.



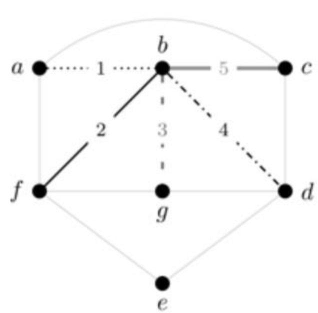
  - ✓ In general, $\chi'(K_n)=n-1$ when n is even and $\chi'(K_n)=n$ when n is odd.
- **Theorem 6.19** (*Vizing's Theorem*) $\Delta(G)\leq\chi'(G)\leq\Delta(G)+1$ for all simple graphs G.
- **Example 6.10** Consider the graph G_4 below and color the edges in the order *ac*, *fg*, *de*, *ef*, *bc*, *cd*, *dg*, *af*, *bd*, *bg*, *bf*, *ab* using a greedy algorithm.

Edge Coloring

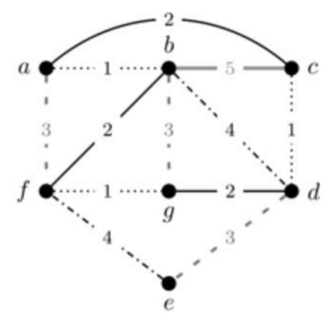


ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab

✓ Another better solution: start edge coloring at the vertex of highest vertex degree



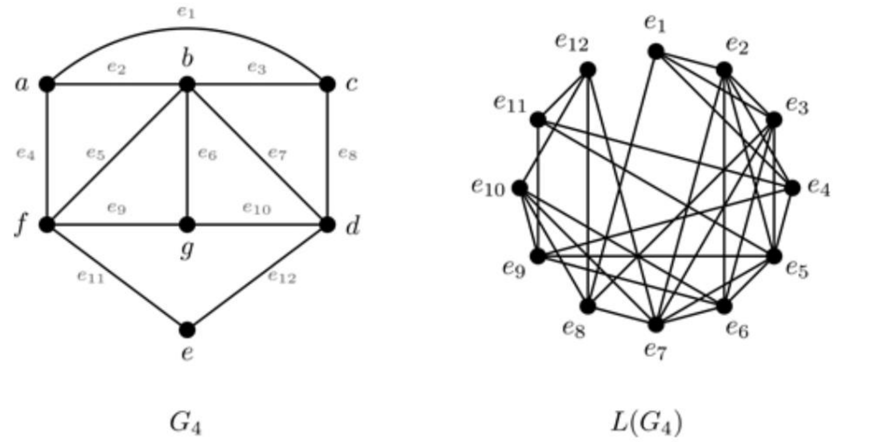
initial coloring of G_4



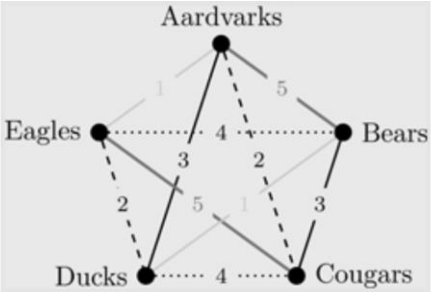
optimal edge-coloring of G_4

Edge Coloring

- **Definition 6.20** Given a graph $G=(V, E)$, the line graph $L(G)=(V', E')$ is the graph formed from G where each vertex x' in $L(G)$ represents the edge x' from G and $x'y'$ is an edge of $L(G)$ if the edges x' and y' share an endpoint in G.

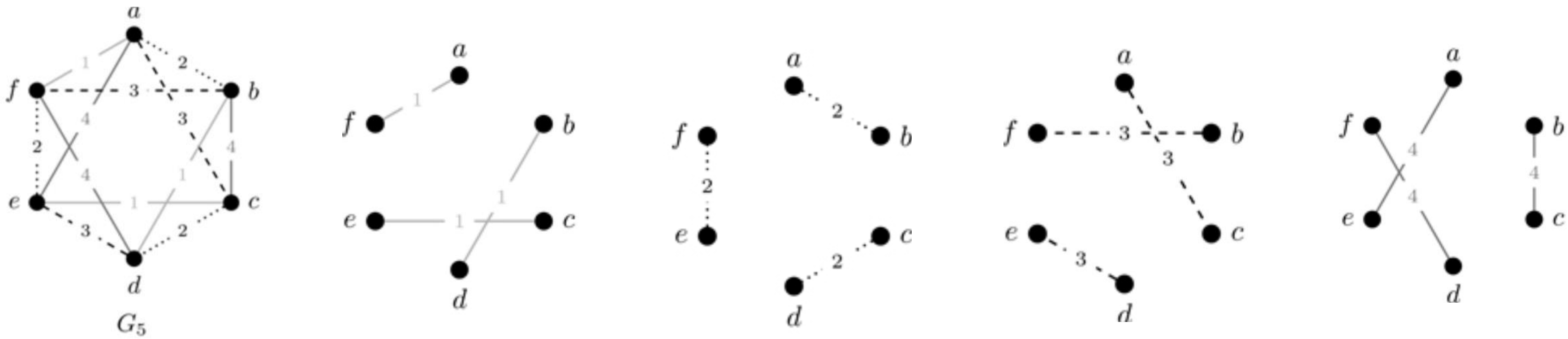


- **Theorem 6.21** Given a graph G with line graph $L(G)$, we have $\chi'(G)=\chi(L(G))$.
- **Example 6.11** The five teams from Section 1.1 need to determine the game schedule for the next year. If each team plays each of the other teams exactly once, determine a schedule where no team plays more than one game on a given weekend.

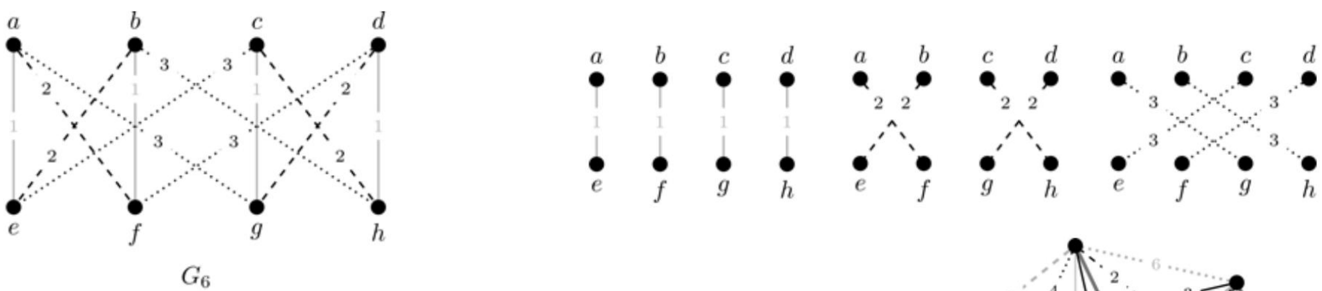


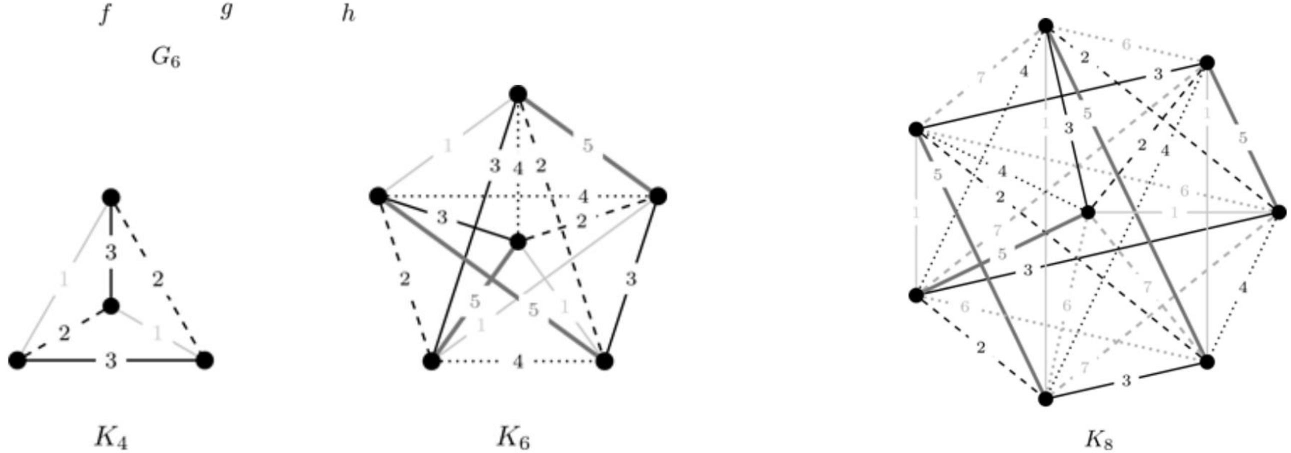
Week	Games	
1	Aardvarks vs. Bears	Cougars vs. Eagles
2	Aardvarks vs. Cougars	Ducks vs. Eagles
3	Aardvarks vs. Ducks	Bears vs. Cougars
4	Aardvarks vs. Eagles	Bears vs. Ducks
5	Bears vs. Eagles	Cougars vs. Ducks

1-Factorizations Revisited



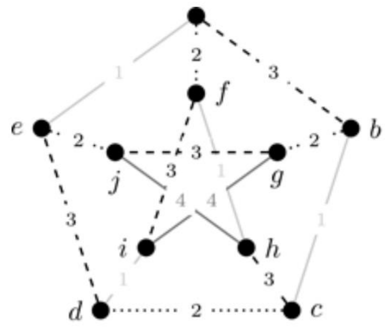
- *Proposition 5.22* states that every *k*-regular bipartite graph has a 1-factorization.





1-Factorizations Revisited

- $K_{2n\text{-}1}$ has no 1-factorizations, neither for Peterson graph (3-reguar but χ'=4)
  - ✓ $2n$-2 regular but $\chi' = 2n$-1

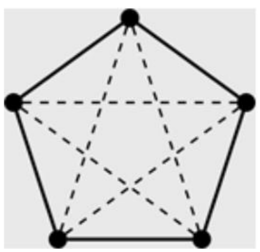


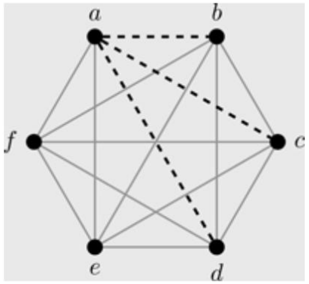
edge-coloring of Petersen graph

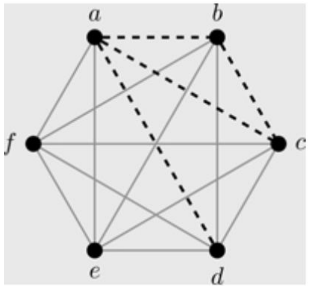
- **Theorem 6.22** Let G be a graph with $2n$ vertices, all of which have degree n if n is odd or $n-1$ if n is even. Then G has a 1-factorization.

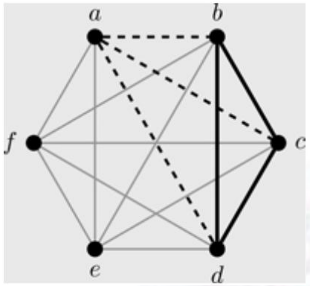
Ramsey Numbers

- **Definition 6.23** Given positive integers m and n, the *Ramsey number* $R(m, n)$ is the minimum number of vertices r so that all simple graphs on r vertices contain either a clique of size m or an independent set of size n.
  - ✓ are often described in terms of guests at a party. For example, if you wanted to find $R(3,2)$, then you would be asking how many guests must be invited so that at least 3 people all know each other or at least 2 people do not know each other.
  - ✓ Proving $R(m, n)=r$ requires two steps: first, we find an edge-coloring of K_{r-1} without a red m-clique and without a blue n-clique; second, we must show that any edge-coloring of K_r will have either a red m-clique or a blue n-clique.
- **Example 6.12** Determine $R(3,3)$.









Ramsey Numbers

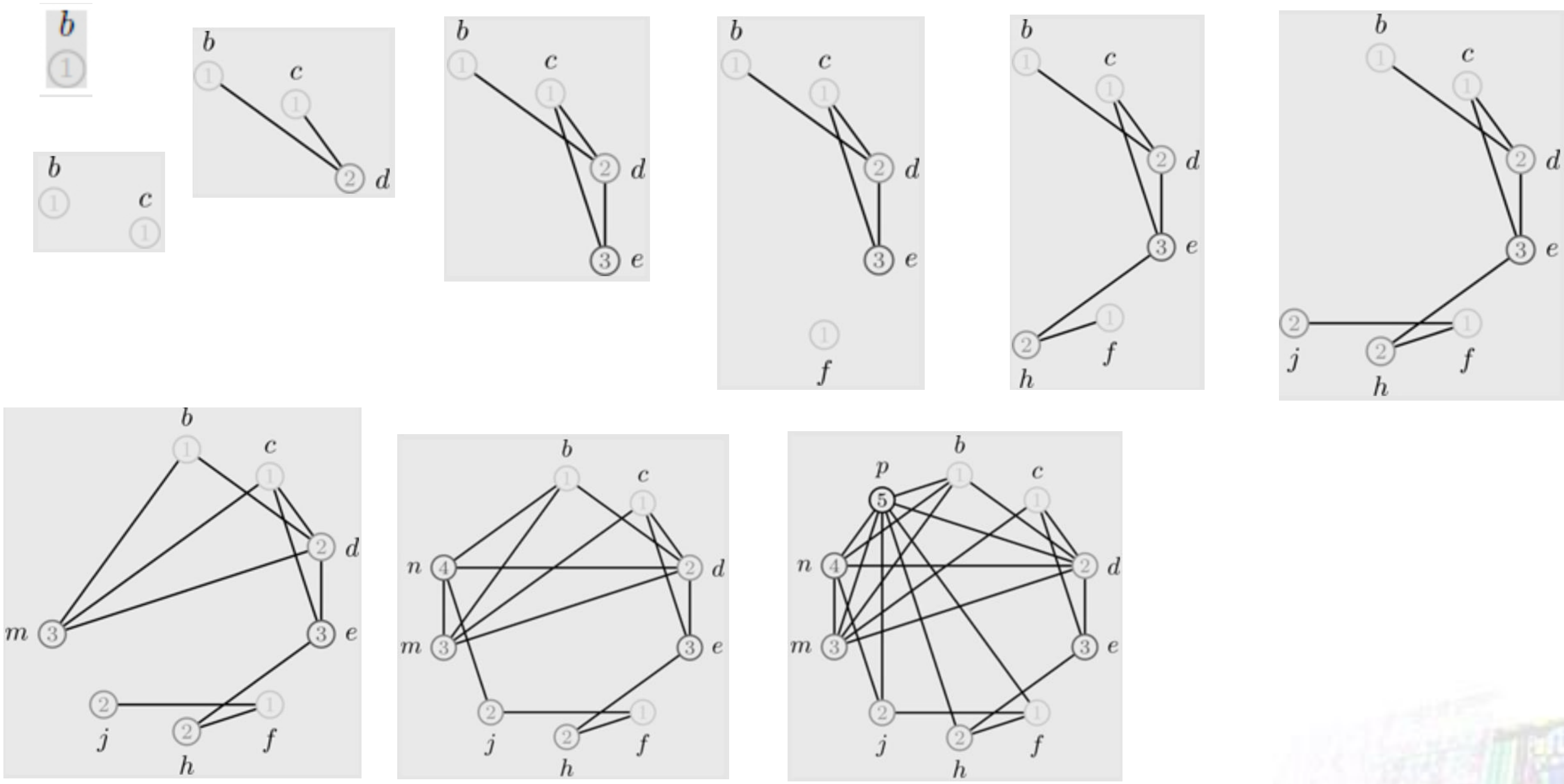
- $R(m, n)=R(n, m)$, $R(2, n)=n$

m / n	3	4	5 lower	5 upper	6 lower	6 upper	7 lower	7 upper
3	6	9	14		18		23	
4		18	25		36	41	49	61
5			43	48	58	87	80	143
6					102	165	115	298
7							205	540

On-Line Coloring

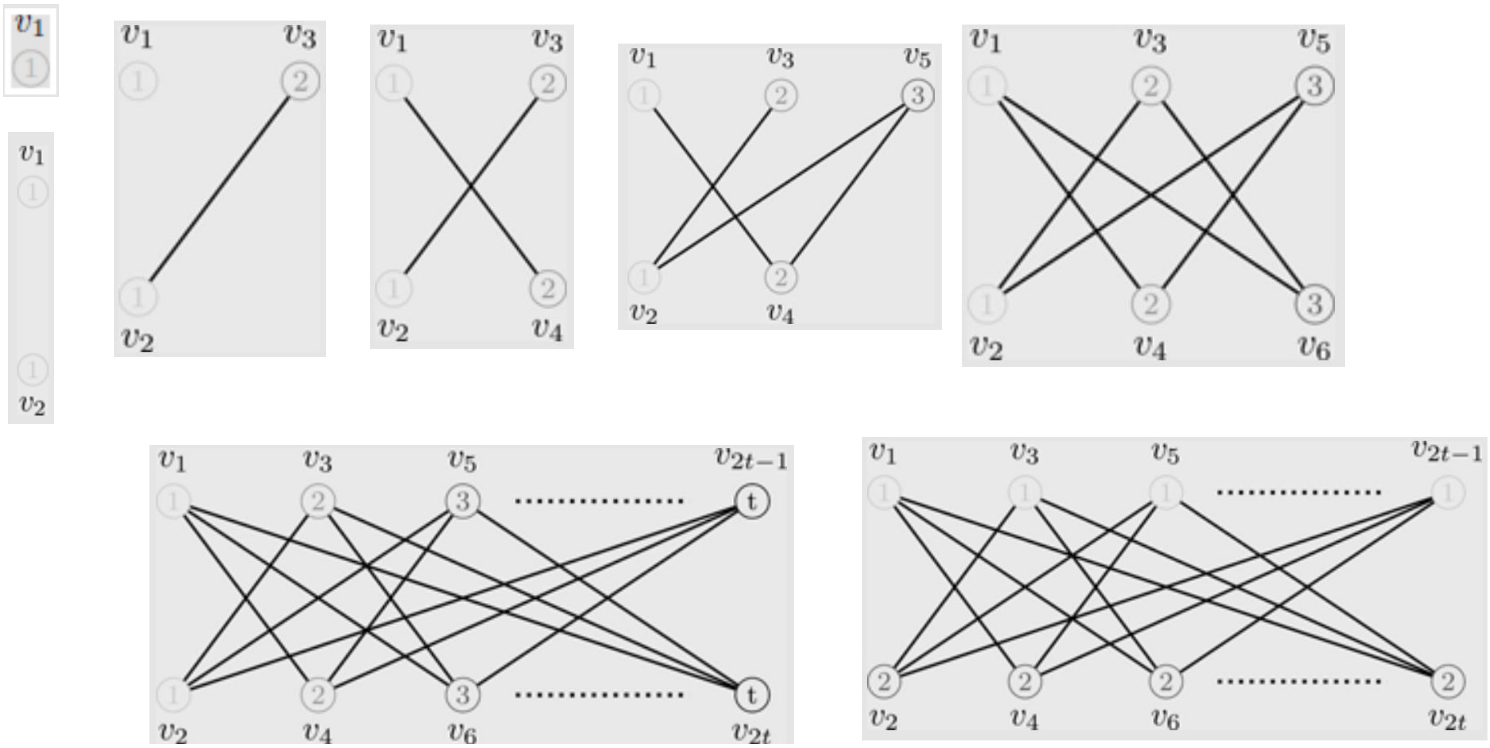
- **Definition 6.24** Consider a graph G with the vertices ordered as $x_1 \prec x_2 \prec \cdots \prec x_n$. An on-line algorithm colors the vertices one at a time where the color for x_i depends on the induced subgraph $G[x, \ldots, x_i]$ which consists of the vertices up to and including x_i. The maximum number of colors a specific algorithm A uses on any possible ordering of the vertices is denoted $\chi A(G)$.
- **First-Fit Coloring Algorithm**
  - *Input*: Graph G with vertices ordered as $x_1 \prec x_2 \prec \cdots \prec x_n$.
  - *Steps*:
    1. Assign x_1 color 1.
    2. Assign x_2 color 1 if x_1 and x_2 are not adjacent; otherwise, assign x_2 color 2.
    3. For all future vertices, assign x_i the least number color available to x_i in $G[x_1, \ldots, x_i]$; that is, give x_i the first color not used by any neighbor of x_i that has already been colored.
  - *Output*: Coloring of G.
- **Example 6.13** Apply the First-Fit Algorithm to the graph from Example 6.3 if the vertices are ordered alphabetically.

On-Line Coloring



- **Example 6.14** A vertex will be revealed one at a time, along with any edges to previously seen vertices. The First-Fit Algorithm will be applied in the order the vertices are seen.

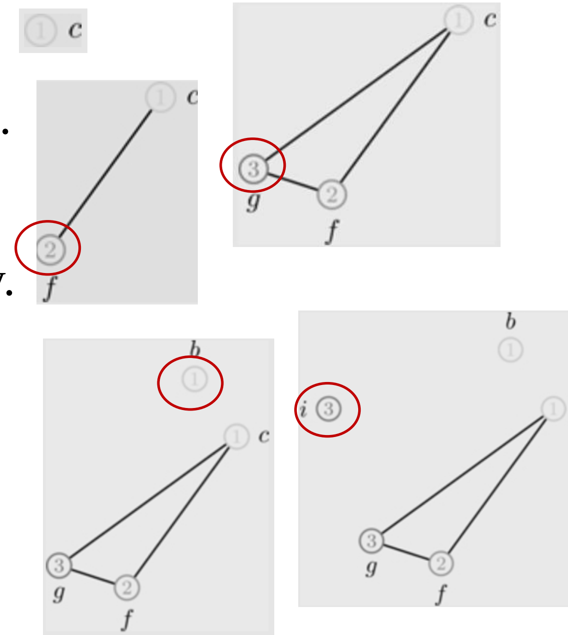
On-Line Coloring

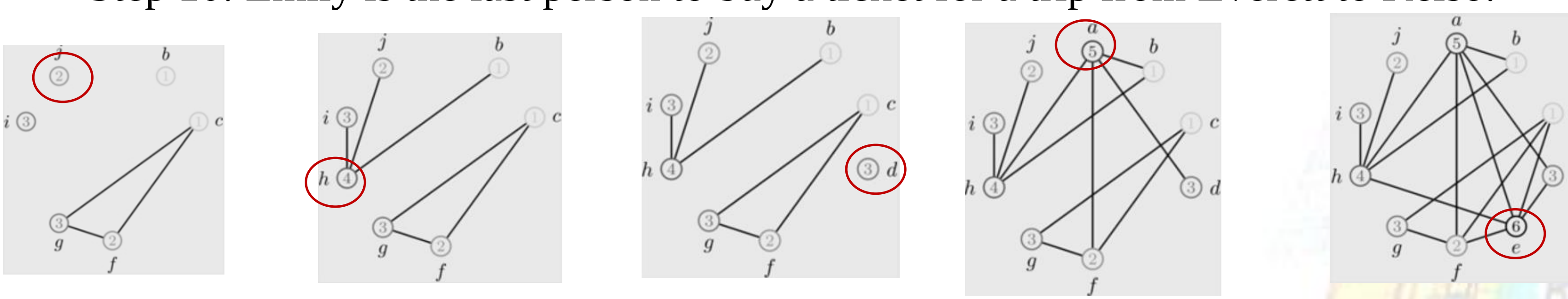


- **Example 6.15** Ten customers are buying tickets for various trips along the Pacific Northwest train route shown on the right. Each person must be assigned a seat when a ticket is purchased and you only know which seats have been previously assigned. Using a *random assignment* of seat numbers as the information becomes available, find a way to minimize the number of seats required.

On-Line Coloring

- ✓ Step 1: Cathy buys her ticket for Bellingham to Edmonds.
- ✓ Step 2: Fiona buys her ticket next for Bellingham to Renton.
- ✓ Step 3: Greg buys a ticket for Vancouver to Mount Vernon.
- ✓ Step 4: Ben buys the next ticket for Portland to Oregon City.
- ✓ Step 5: Next Ingrid buys a ticket for Oregon City to Albany.
- ✓ Step 6: Jessica buys the next ticket for Albany to Eugene.
- ✓ Step 7: Howard buys a ticket for Centralia to Eugene.
- ✓ Step 8: Dana buys the next ticket for Renton to Tacoma.
- ✓ Step 9: Aiden buys a ticket for Seattle to Oregon City.
- ✓ Step 10: Emily is the last person to buy a ticket for a trip from Everett to Kelso.

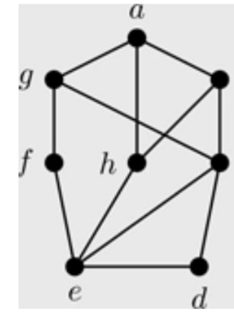




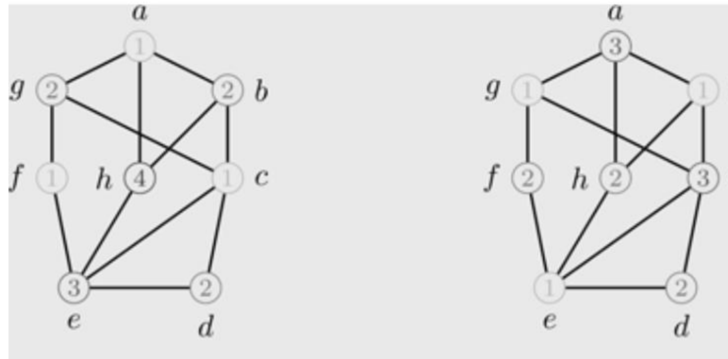
Vancouver - Bellingham - Mount Vermon - Stanwood - Everett - Edmonds - Seattle - Renton - Tacoma - Olympia-Lacey – Centralla - Kelso - Portland - Oregon City - Salem - Ajbany - Eugene

On-Line Coloring

- **Example 6.16** Color the graph below using First-Fit using the two different vertex orders listed and determine if either finds the optimal coloring for the graph.



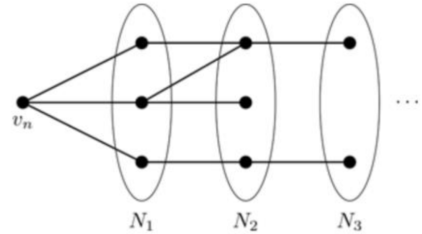
  - ✓ a≺b≺c≺d≺e≺f≺g≺h
  - ✓ e≺h≺b≺d≺c≺g≺f≺a



vertex order (a) vertex order (b)

Proof of Brooks' Theorem

- **Theorem 6.7** (*Brooks' Theorem*) Let G be a connected graph and Δ denote the maximum degree among all vertices in G. Then $\chi(G) \le \Delta$ as long as G is not a complete graph or an odd cycle. If G is a complete graph or an odd cycle then $\chi(G)=\Delta+1$.
  - ✓ Assume G is a connected graph that is not complete nor an odd cycle. Let $k=\Delta(G)$. If $k=0$ or $k=1$ then G must be complete. If $k=2$ then G is either a path or an even cycle $\rightarrow$ 2 colors.
  - ✓ $k \ge 3$. Color G using at most k colors with the vertex as $v_1 \prec v_2 \prec \cdots \prec v_n$ and $\forall\ i$, v_i has at most $k-1$ neighbors preceding it in the list $\rightarrow$ *First-Fit* will use at most k colors.



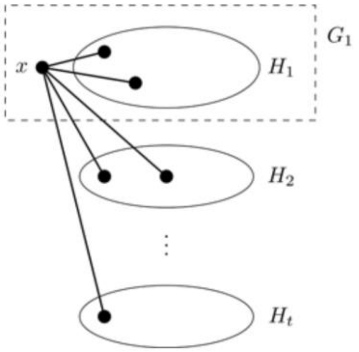
  - ✓ **Case 1: G is not regular**.
    - ➢ $\exists v_n$, $deg(v_n) \le (k-1)$. Let N_i be the set of vertices of distance i from v_n; that is $N_0=\{v_n\}, N_1=N(v_n)$ and N_2 is the set of vertices whose shortest path to v_n is of length 2, and so on.
    - ➢ Since G is finite, there must be some value t for which N_1 is the last set created. We will label the vertices in reverse order based on the N_i for which they belong.

$$\underbrace{v_1 \prec v_2 \prec \cdots}_{N_t} \prec \underbrace{v_i \prec v_{i+1} \prec \cdots}_{N_{t-1}} \prec \cdots \prec \underbrace{v_j \prec \ldots \prec v_{n-1}}_{N_1} \prec v_n$$

    - ➢ Note that every vertex has at most $k-1$ neighbors preceding it in the list. Thus First-Fit will use at most k colors on vertices $v_1, v_2, \ldots, v_{n-1}$ and since v_n has degree at most $k-1$.

Proof of Brooks' Theorem

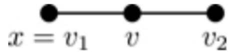
- **Case 2: G is regular and G has a cut vertex, call it x.**
  - Then every vertex has degree k and $G-x$ must have at least two components, call them H_1, $H_2,\ldots, H_t$, where $t\geq 2$.
  - Let $G_i=H_i\cup\{x\}$, that is G_i is the graph created by putting x back into H_i along with its edges to the vertices in H_i.
  - Then x must have degree at most $k-1$ in each G_i, and so we can use the method from Case 1 to properly color each G_i with at most k colors.
  - Permute the colors of G_i ,if necessary, such that the color of x in each G_i is the same.



- **Case 3: G is regular and does not contain a cut vertex.**
  - Then G must be 2-connected. Note that in any vertex ordering v_n will have k earlier neighbors.
  - We want to show that two of these neighbors are given the same color.

Proof of Brooks' Theorem

- Within each of the subcases below we will identify vertices v, v_1, and v_2 so that $vv_1, vv_2 \in E(G)$ but $v_1v_2 \notin E(G)$ and $G-\{v_1, v_2\}$ is still connected. Let x be an arbitrarily chosen vertex of G.
- **Case 3a**: **Suppose $G-x$ is 2-connected**.



Since G is not complete, $\exists v_2$, $dist(x, v_2) = 2$. Let $x = v_1$ and choose v to be the common neighbor of x and v_2. Then $vv_1, vv_2 \in E(G)$ but $v_1v_2 \notin E(G)$.

$G-\{v_1, v_2\}$ must still be connected since $G-x=G-v_1$ is 2-connected and so v_2 cannot be a cut-vertex in $G-x$.

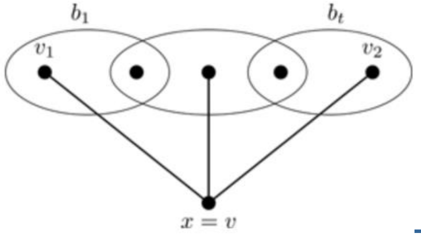
- **Case 3b**: **Suppose $G-x$ is not 2-connected**.

Then $G-x$ is not itself a block, and so must contain blocks $b_1, b_2, \ldots, b_t$.

Since the block graph $B(G-x)$ of $G-x$ is a tree (see Exercise 4.30), we know at least two blocks, say b_1 and b_t, are leaves in $B(G-x) \rightarrow$ both b_1 and b_t contain cut vertices of $G-x$.

If either of these cut-vertices c were the only neighbor of x in the block, then $G-c$ would be disconnected $\rightarrow$ contradiction to G-x is connected.

Let $v=x$ and v_1 and v_2 be a vertex from b_1 and b_t, respectively, that is not a cut-vertex of $G-x$. Then v_1 and v_2 are not adjacent and since $deg(x) \geq 3$, we know $G-\{v_1, v_2\}$ is connected.

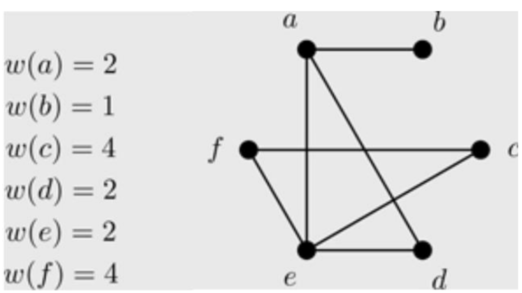


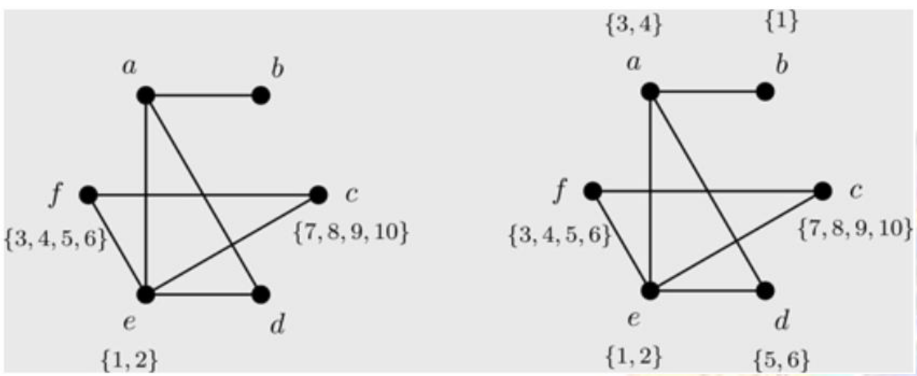
Proof of Brooks' Theorem

- Order the vertices of G, with v_1 and v_2 identified as above and $v=v_n$.
- For $v_{n\text{-}1}$, choose a vertex that is adjacent to v that is neither v_1 nor v_2 (which we know exists since $deg(v_n \geq 3$).
- For $v_{n\text{-}2}$ choose a vertex adjacent to either v_n or $v_{n\text{-}1}$ that hasn't already been chosen.
- Note that since $G-\{v_1, v_2\}$ is connected, for each $i \in \{3,\ldots,n-1\}$ there is an available vertex $v_i \in V(G)-\{v_1, v_2, v_n, v_{n\text{-}1},\ldots, v_{i+1}\}$ that is adjacent to at least one of $v_{i+1},\ldots, v_n$.
- Continue choosing vertices in this way until all vertices have been labeled.
- Now using First-Fit on this ordering of the vertices, and assign color 1 for v_1 and v_2.
- Since every vertex v_i, for $3 \leq i < n$ has a neighbor after it in the order, we know it is adjacent to at most $k-1$ vertices preceding it. Thus First-Fit will need at most k colors for G.

Weighted Coloring

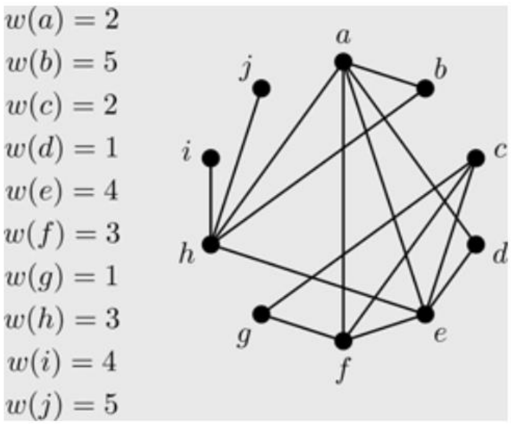
- Consider the following scenario: Ten families need to buy train tickets for an upcoming trip. The families vary in size but each of them needs to sit together on the train. Determine the minimum number of seats needed to accommodate the ten family trips.
- **Definition 6.25** Given a *weighted graph* G=(V, E, w), where w assigns each vertex a positive integer, a *proper weighted coloring* of G assigns each vertex a set of colors so that
  - ✓ the set consists of consecutive colors (or numbers);
  - ✓ the number of colors assigned to a vertex equals its weight;
  - ✓ if two vertices are adjacent, then their set of colors must be disjoint.
- **Example 6.18** Find an optimal weighted coloring for the graph below where the vertices have weights as shown below.

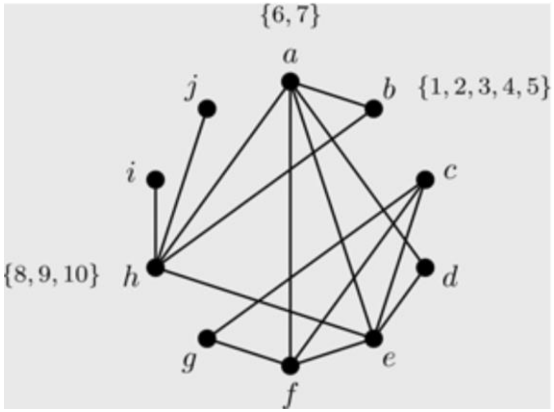


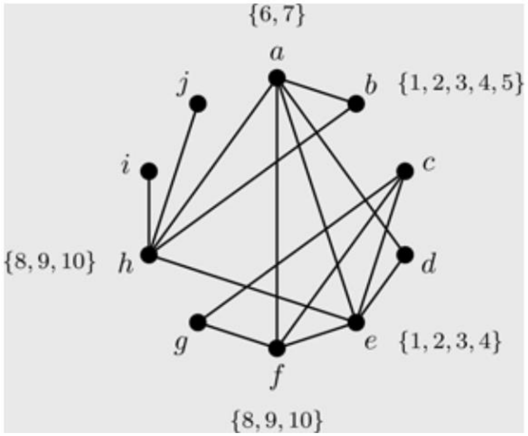


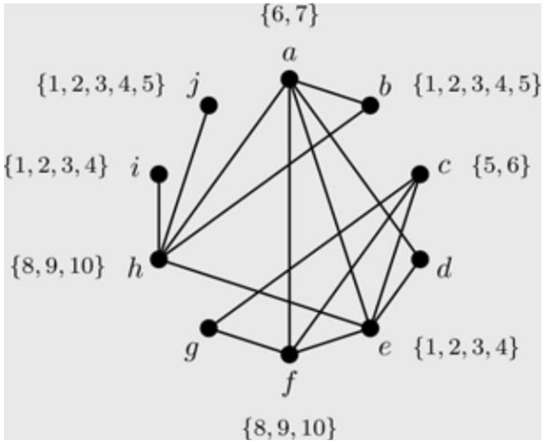
Weighted Coloring

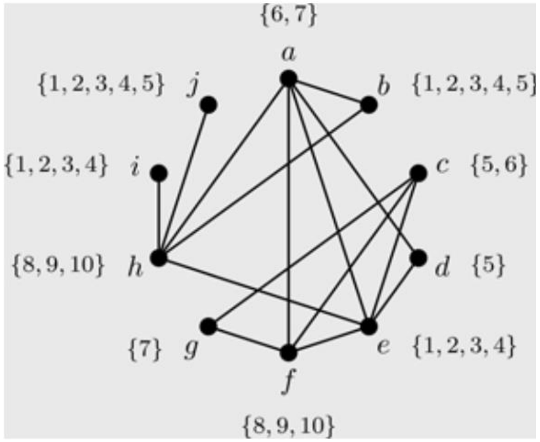
- **Example 6.19** Suppose the ten families needing train tickets have the same underlying graph as that from Example 6.15 and the size of each family is noted below. Determine the minimum number of seats needed to accommodate everyone's travels.







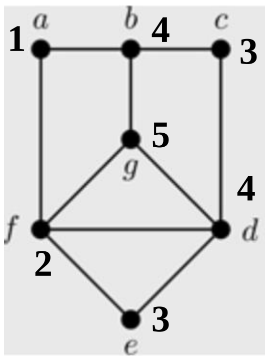




- Application: Dynamic storage allocation (DSA) – assign sized variables to memory locations

List Coloring

- **Example 6.20** The table on the right below shows two different lists for the vertices in the graph on the left. Find a proper coloring for each version of the lists or explain why none exists.
  - ✓ A possible coloring from the first set of lists can be given as *a*–1, *b*–4, *c*–3, *d*–4, *e*–3, *f*–2, *g*–5.



Vertex	List 1	List 2
a	$\{1,2\}$	$\{1\}$
b	$\{1,4,5\}$	$\{1,3,5\}$
c	$\{3,4\}$	$\{3,4\}$
d	$\{3,4\}$	$\{3,4\}$
e	$\{2,3\}$	$\{2,3\}$
f	$\{1,2,3\}$	$\{1,2,3\}$
g	$\{4,5\}$	$\{4,5\}$

- **Definition 6.26** Let *G* be a graph where each vertex *x* is given a list *L*(*x*) of colors. A *proper list coloring* assigns to *x* a color from its list *L*(*x*) so that no two adjacent vertices are given the same color.
  - ✓ If for every collection of lists, each of size *k*, a proper list coloring exists then *G* is *k-choosable*. The minimum value for *k* for which *G* is *k*-choosable is called the *choosability* of *G* and denoted *ch*(*G*).

List Coloring

- **Proposition 6.27** For any simple graph G, $ch(G) \geq \chi(G)$.
  - ✓ Let G satisfy $\chi(G)=k$ and give each vertex of G the list $\{1, 2, \ldots, k\}$. Then there is a proper coloring for G from these lists, namely the one exhibited by the fact that $\chi(G)=k$.
  - ✓ However, if we remove the same one element from each of these lists, then G cannot be colored since otherwise $\chi(G)<k$.
- **Proposition 6.28** For any simple graph G, $ch(G) \leq \Delta(G)+1$.