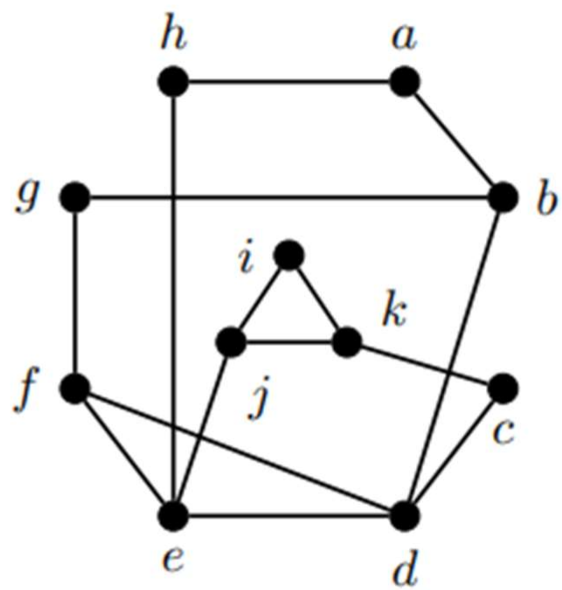
National University
of computer and emerging sciences

Foundation for Advancement of Science and Technology

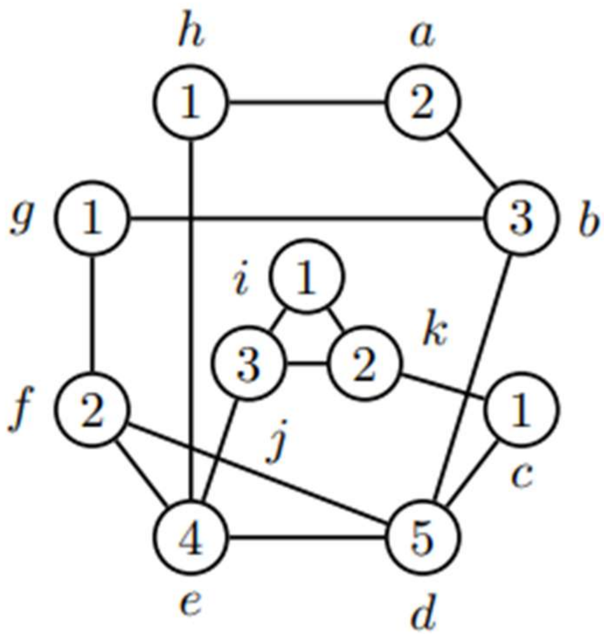
Lec # 34, 35 & 36

Graph Coloring

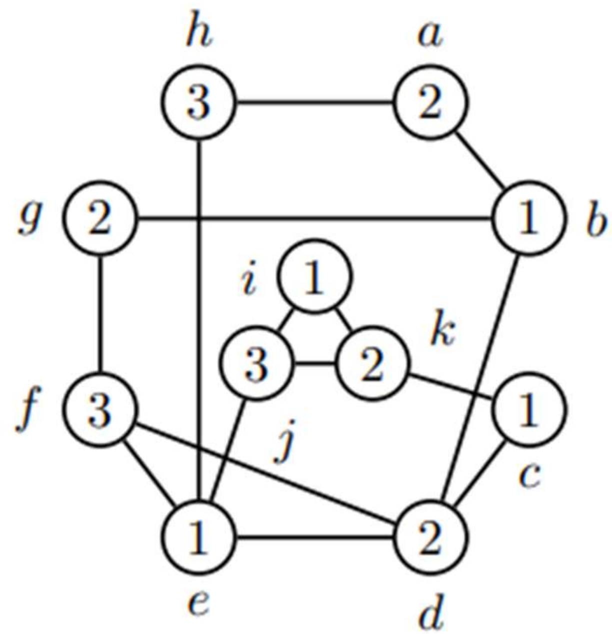
Definition 6.1 A proper ***k-coloring*** of a graph G is an assignment of colors to the vertices of G so that no two adjacent vertices are given the same color and exactly k colors are used.



G_1



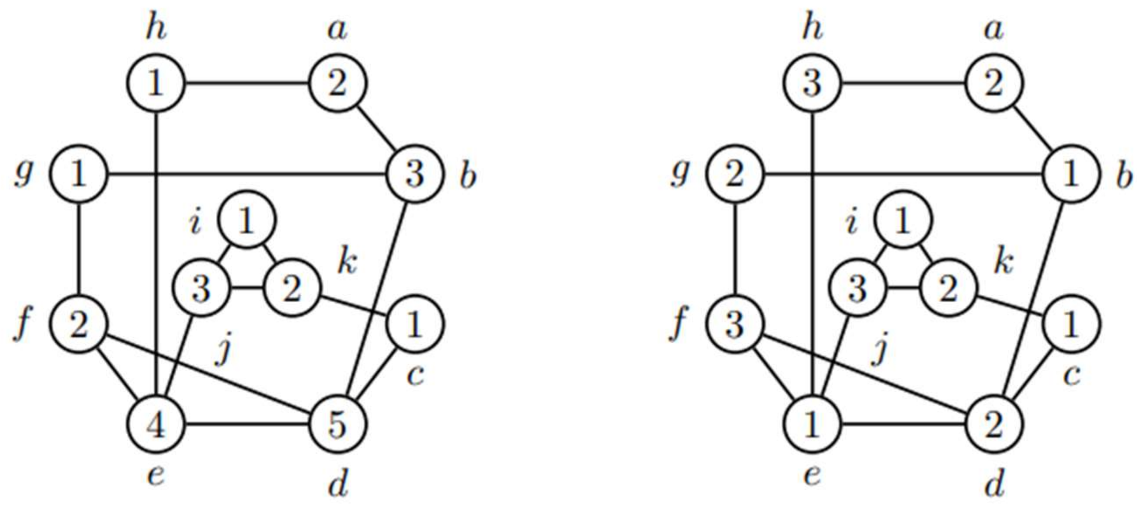
proper 5-coloring



proper 3-coloring

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.

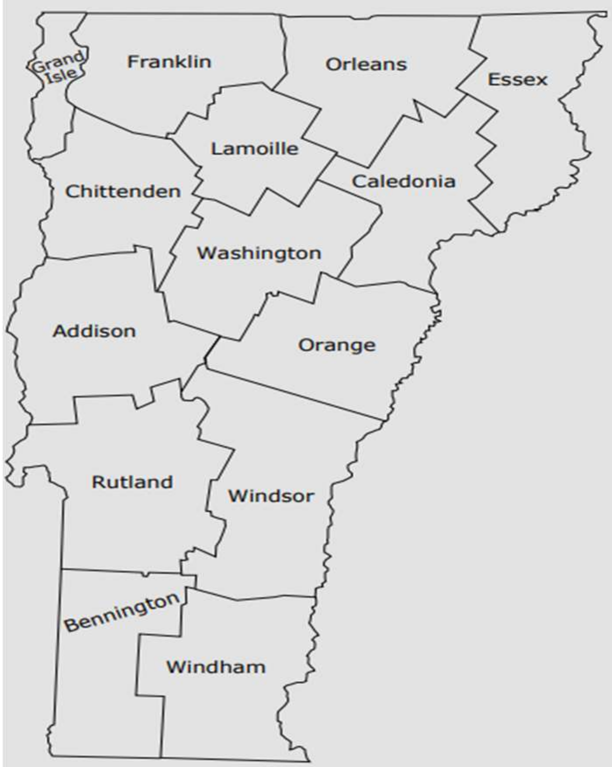
Definition 6.3 The ***independence number*** of a graph G is $\alpha(G) = n$ if there exists a set of n vertices with no edges between them but every set of $n + 1$ vertices contains at least one edge.

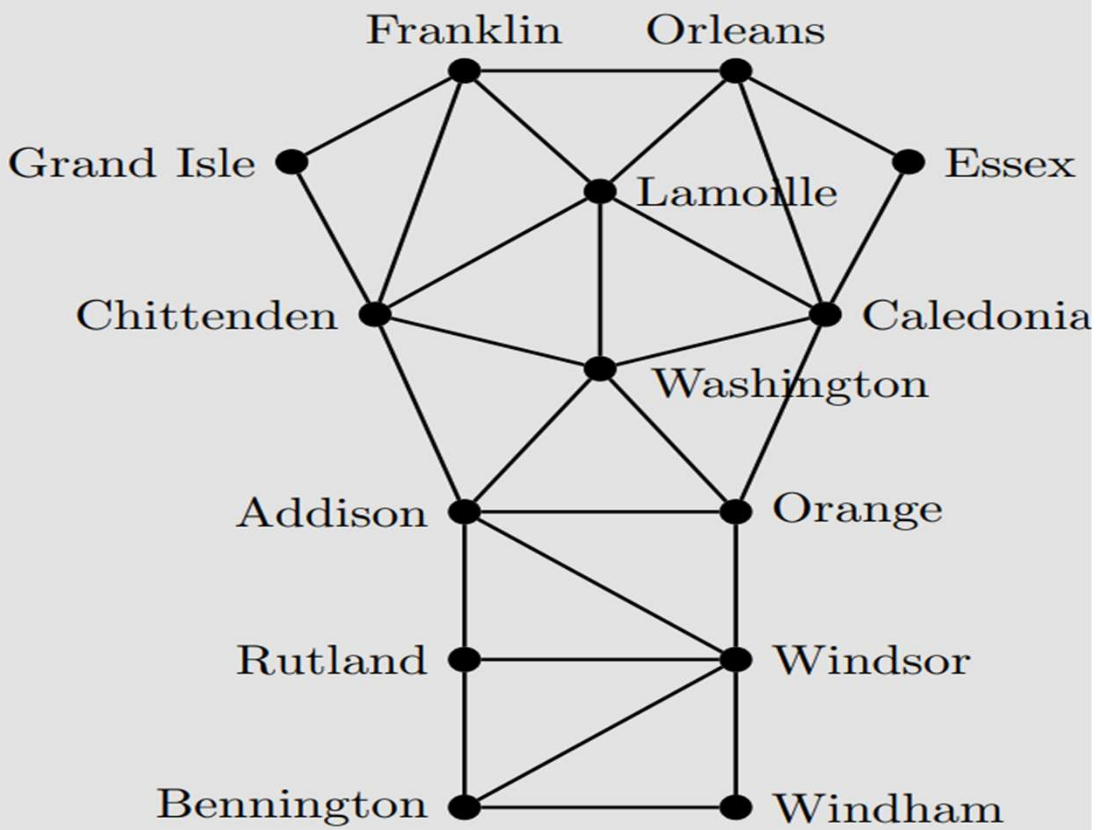


proper 5-coloring　　　　proper 3-coloring

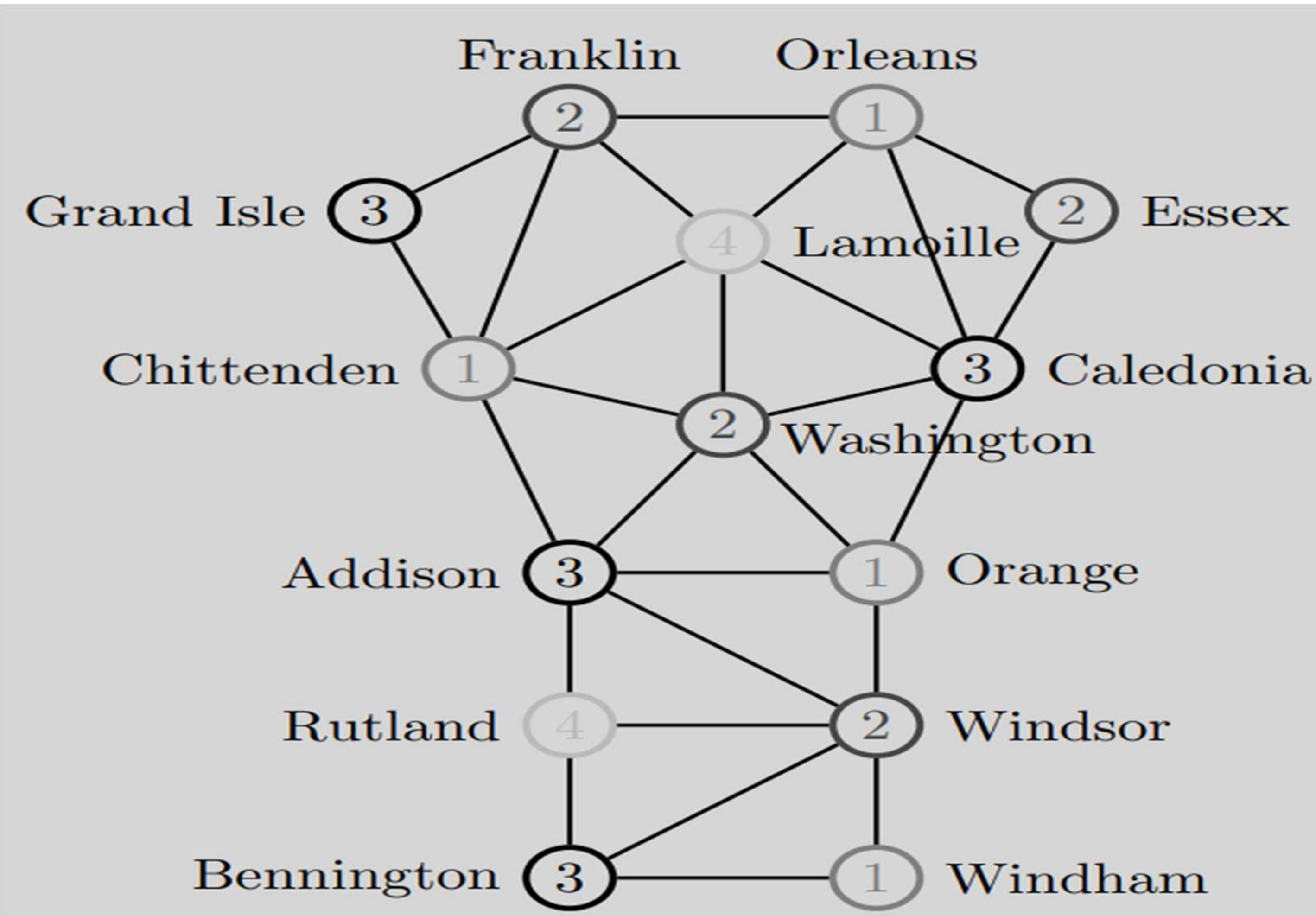
The first coloring of G_1 has color classes $S_1 = \{c, g, h, i\}$, $S_2 = \{a, f, k\}$, $S_3 = \{b, j\}, S_4 = \{e\}, S_5 = \{d\}$ and the second coloring has color classes $S_1 = \{b, c, e, i\}, S_2 = \{a, d, g, k\}, S_3 = \{f, h, j\}$. Recall that two vertices are independent if they have no edges between them. Thus a color class must consist of independent vertices. We will see there is a relationship between independent sets and coloring a graph.

Example 6.1 Find a coloring of the map of the counties of Vermont and explain why three colors will not suffice.

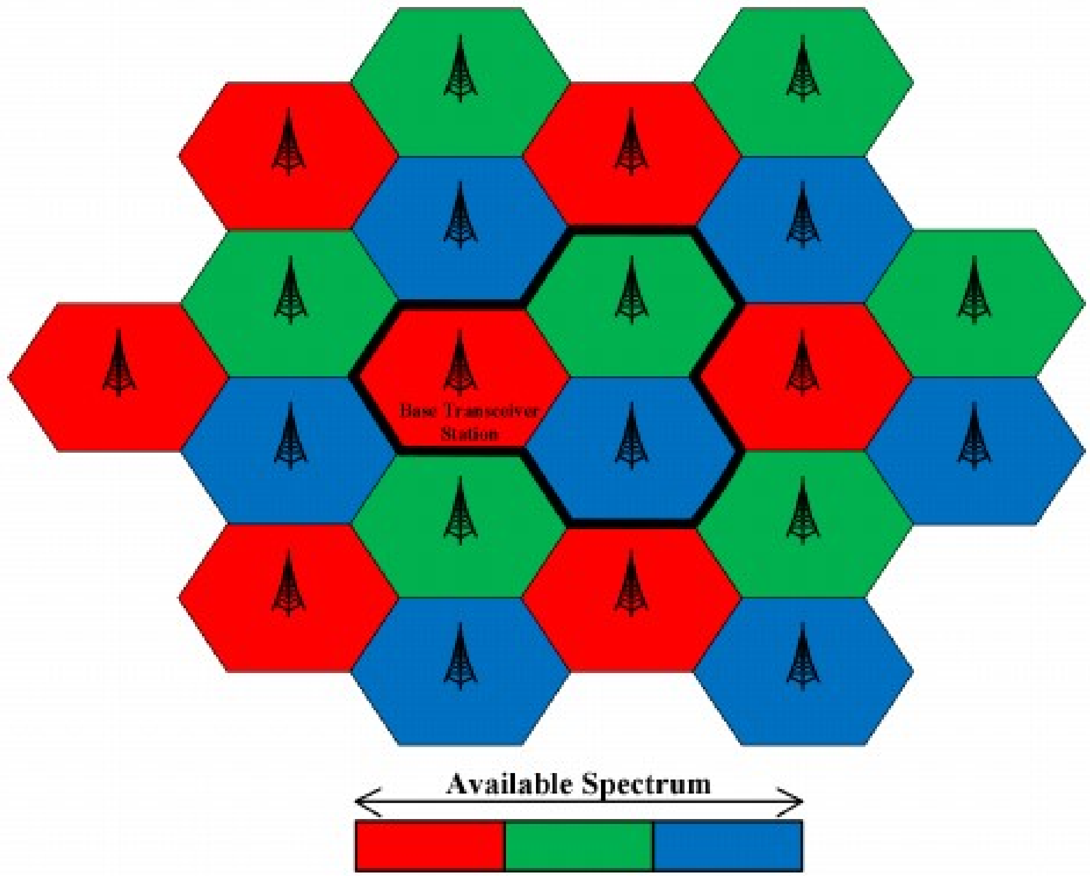




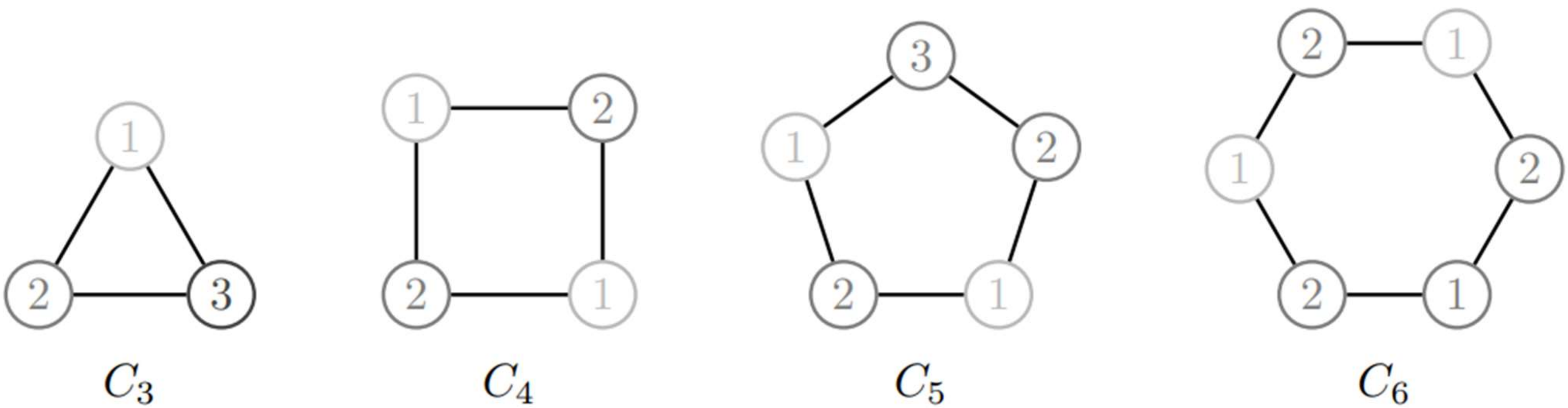
National University of computer and emerging sciences



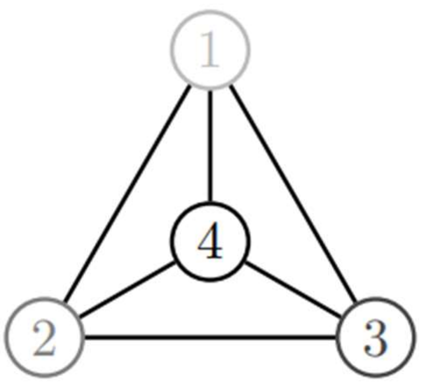
Note that Lamoille County is surrounded by five other counties. If we try to alternate colors amongst these five counties, for example Orleans – 1, Franklin – 2, Chittenden – 1, Washington – 2, we still need a third color for the fifth county (Caledonia – 3). Since Lamoille touches each of these counties, we know it needs a fourth color.


Base Transceiver Station
Available Spectrum

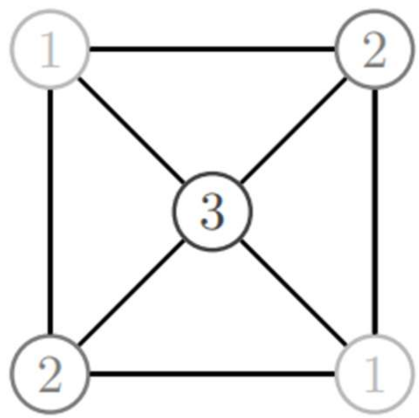
Definition 6.4 The ***chromatic number*** $\chi(G)$ of a graph is the smallest value k for which G has a proper k-coloring.



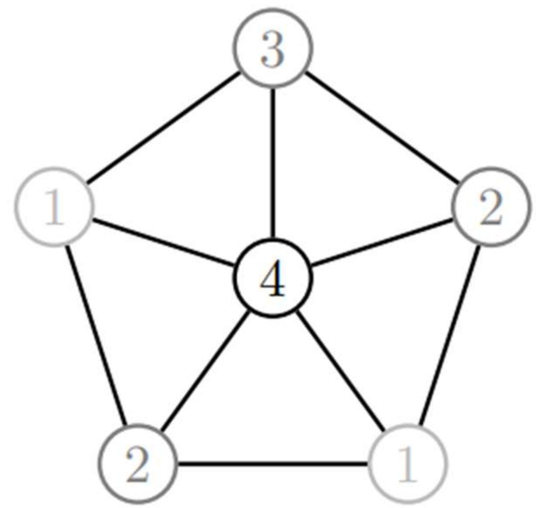
Definition 6.5 A ***wheel*** W_n is a graph in which n vertices form a cycle around a central vertex that is adjacent to each of the vertices in the cycle.



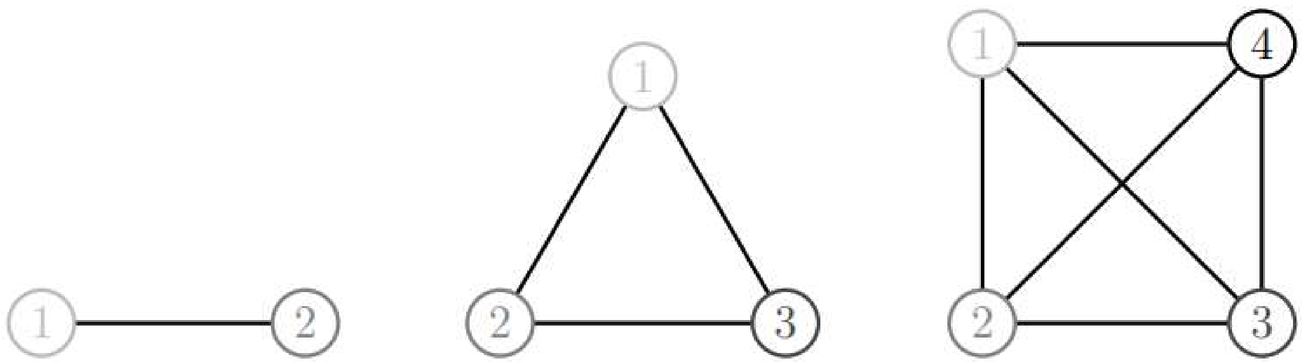
W_3



W_4



W_5



K_2 K_3 K_4

When a complete graph appears as a subgraph within a larger graph, we call it a *clique*.

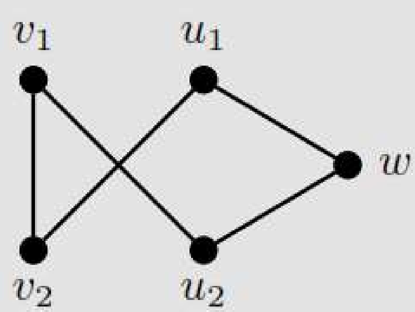
Special Classes of Graphs with known $\chi(G)$

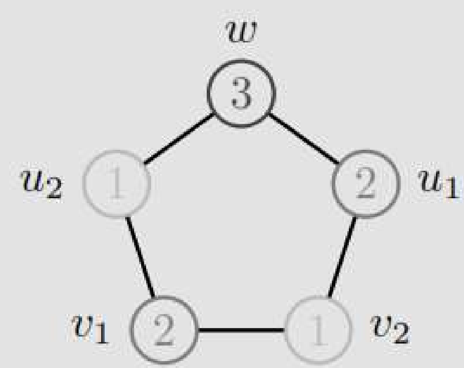
- $\chi(C_n) = 2$ if n is even $(n \geq 2)$
- $\chi(C_n) = 3$ if n is odd $(n \geq 3)$
- $\chi(K_n) = n$
- $\chi(W_n) = 4$ if n is odd $(n \geq 3)$

Example 6.2 Mycielski's Construction is a well-known procedure in graph theory that produces triangle-free graphs with increasing chromatic numbers. The idea is to begin with a triangle-free graph G where $V(G) = \{v_1, v_2, \ldots, v_n\}$ and 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. In addition, we add a new vertex w so that $N(w) = U$; that is, add an edge from w to every vertex in U. The resulting graph will remain triangle-free but need one more color than G. If you perform enough iterations of this procedure, you can obtain a graph with $\omega(G) = 2$ and $\chi(G) = k$ for any desired value of k.

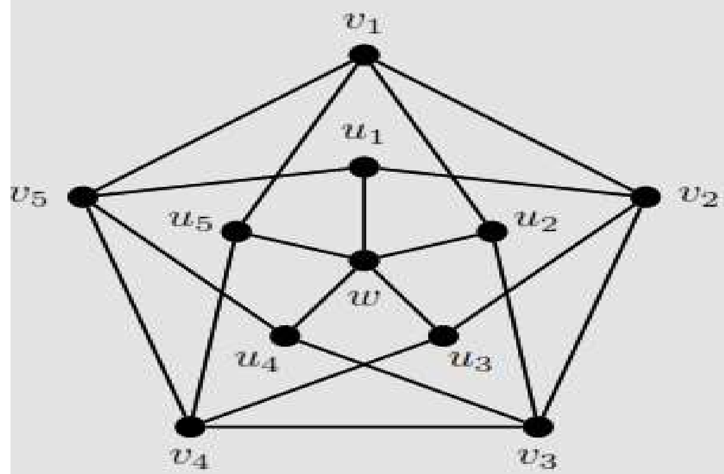
Consider G to be the complete graph on two vertices, K_2, which is clearly triangle free and has chromatic number 2, as shown in the following graph.

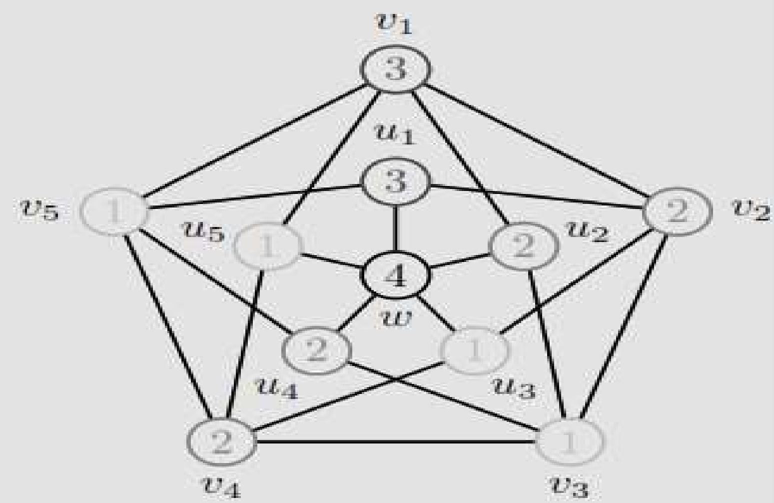
After the first iteration of Mycielski's Construction, we get the graph shown below on the left. Notice that u_1 has an edge to v_2 since v_1 is adjacent to v_2. Similarly, u_2 has an edge to v_1. In addition, w is adjacent to both u_1 and u_2. The graph on the right below is an unraveling of the graph on the left. Thus we have obtained C_5, which we know needs 3 colors.





After the second iteration, we obtain the graph shown below. The outer cycle on 5 vertices represents the graph obtained above in the first iteration. The inner vertices are the new additions to the graph, with u_1 adjacent to v_2 and v_5 since v_1 is adjacent to v_2 and v_5. Similar arguments hold for the remaining u-vertices and the center vertex w is adjacent to all of the u-vertices. A coloring of the graph is shown below on the right. Note that the outer cycle needs 3 colors, as does the group of u-vertices. This forces w to use a fourth color. In addition, no matter which three vertices you choose, you cannot find a triangle among them, and so the graph remains triangle-free.





If we continue this procedure through one more step, we obtain a graph needing 5 colors with a clique size of 2.

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 $\chi(G) = \Delta + 1$.

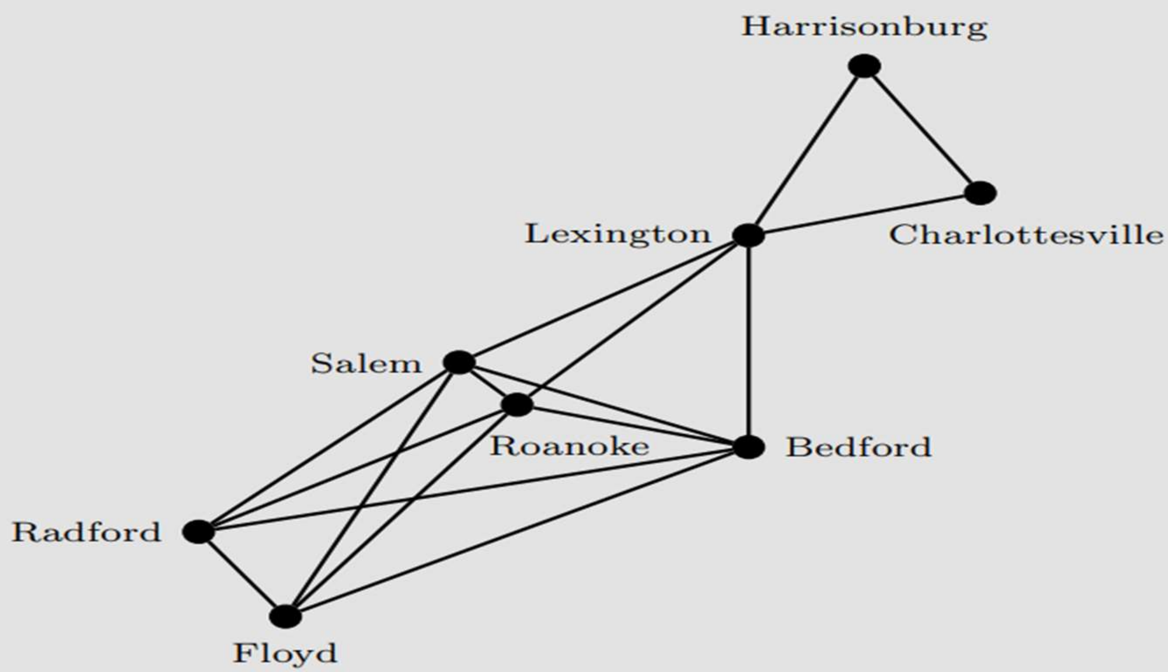
Prove Theorem 9.1: $\chi(G) \leq \Delta(G) + 1$ for any graph G.

Solution. Let n be number of vertices of G. The proof is by induction on n. Let G' be the graph obtained by deleting any vertex v from G. Then $\Delta(G') \leq \Delta(G)$. By the induction hypothesis, $\chi(G') \leq \Delta(G') + 1 \leq \Delta(G) + 1$. We can assign a color to v (in G) that is different from the colors (assigned in G') to the vertices adjacent to it. In that case, we do not need more than $\Delta(G) + 1$ colors to color the vertices of G since the number of vertices adjacent to v is at most $\Delta(G)$. Hence, $\chi(G) \leq \Delta(G) + 1$.

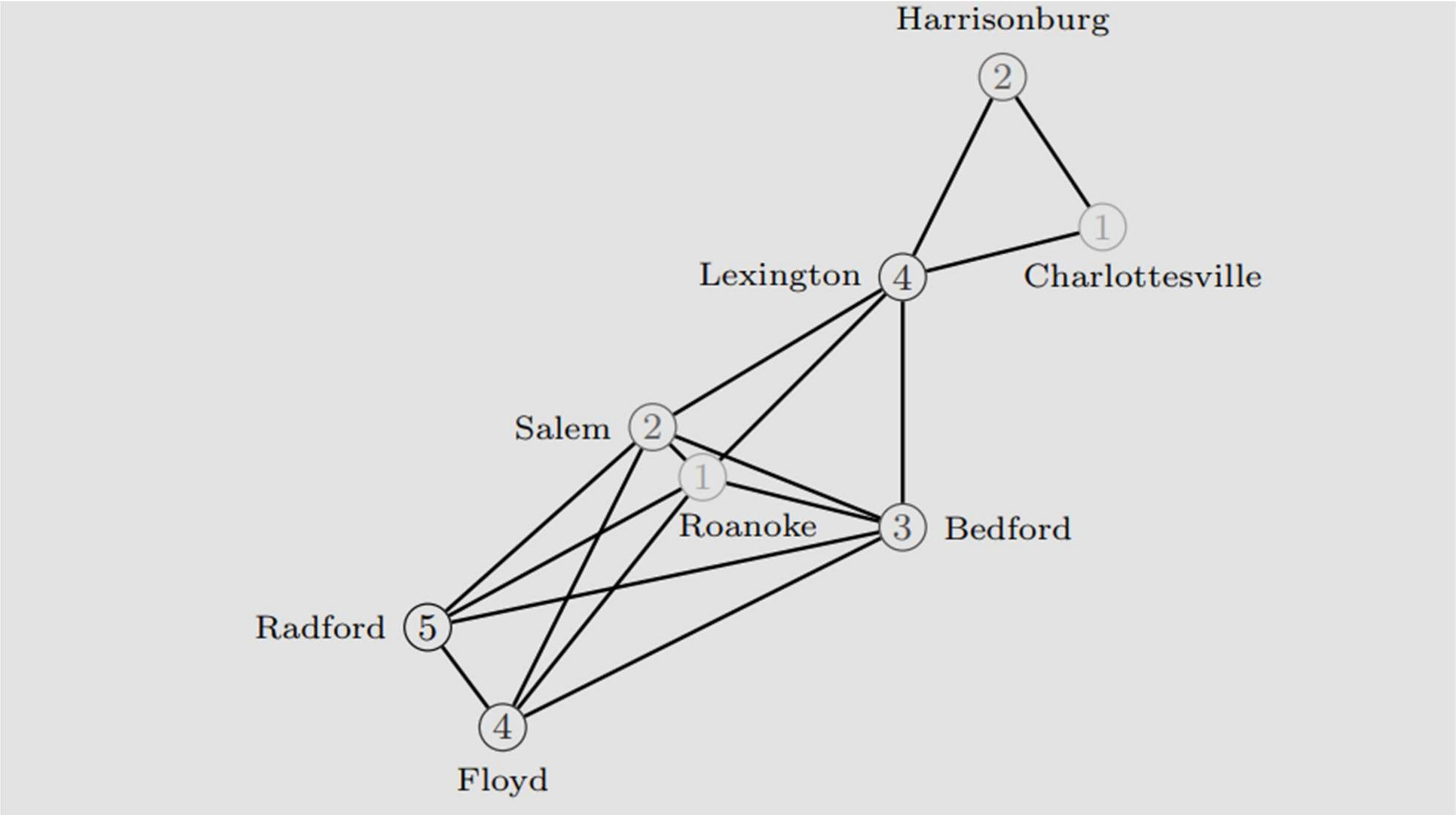
Basic Coloring Strategies

- 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. Determine the fewest number of frequencies need for each city shown below (not drawn to scale) to have its own municipal radio station.



Solution: Each vertex will be assigned a color that corresponds to a radio frequency. This graph has $\chi = 5$ since we have a 5-coloring, as shown below, and fewer than 5 colors will not suffice as there is a K_5 among the vertices representing the cities of Roanoke, Salem, Bedford, Floyd, and Radford.


Harrisonburg
2
1
Lexington
4
Charlottesville
Salem
2
1
Roanoke
3
Bedford
Radford
5
4
Floyd

Proposition 6.9 Let G be a graph with m edges. Then

$$\chi(G) \leq \frac{1}{2} + \sqrt{2m + \frac{1}{4}}$$

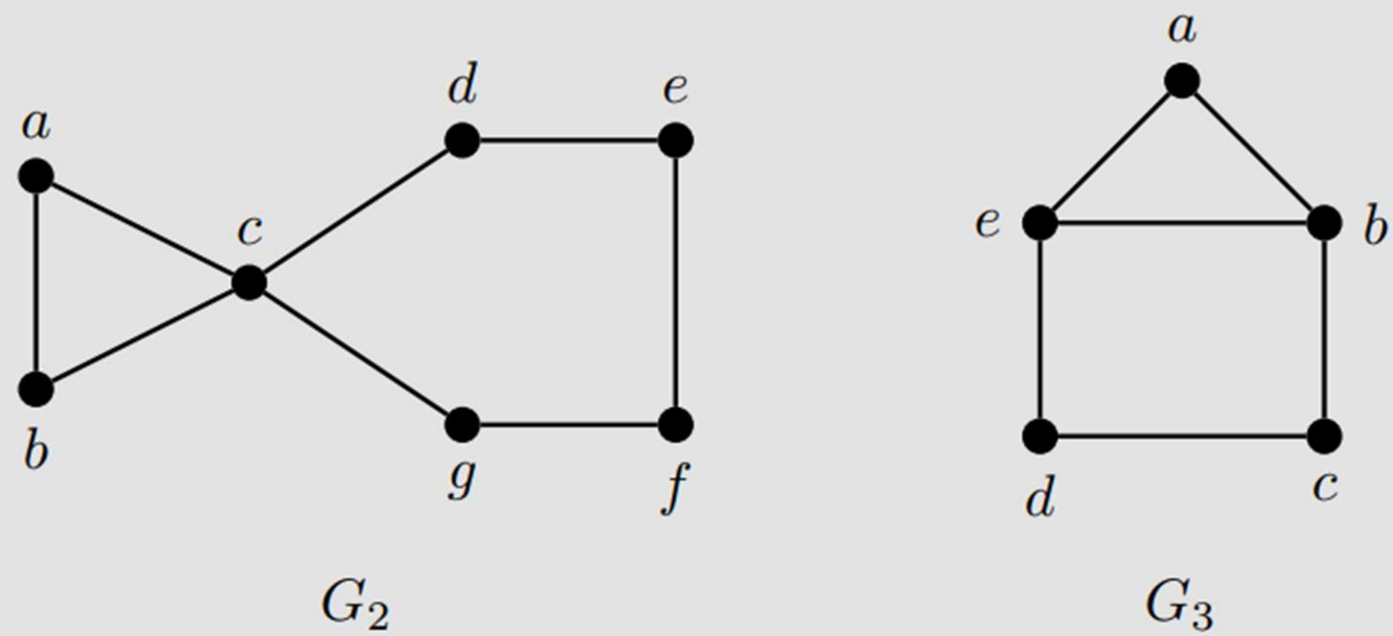
Proposition 6.10 Let G be a graph and $l(G)$ be the length of the longest path in G. Then $\chi(G) \leq 1 + l(G)$.

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.

Definition 6.13 A graph G is ***perfect*** if and only if $\chi(H) = \omega(H)$ for all induced subgraphs H.

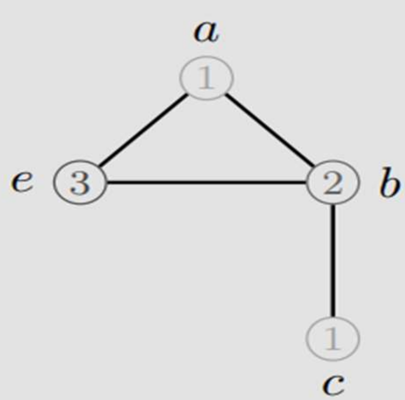
Definition 1.5 Given a graph $G = (V, E)$, an ***induced subgraph*** is a subgraph $G[V']$ where $V' \subseteq V$ and every available edge from G between the vertices in V' is included.

Example 6.6 Determine if either of the two graphs below are perfect.

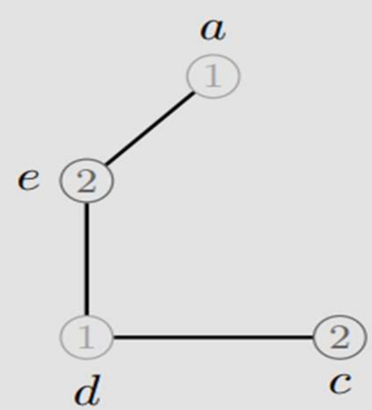


Solution: Without much work we can see that both graphs above satisfy $\chi(G) = \omega(G)$. However, if we look at the subgraph H induced by $\{c, d, e, f, g\}$ in G_2 we see that H is just C_5 and so $\chi(H) = 3$ even though $\omega(H) = 2$. Thus G_2 is not perfect.

However, G_3 is in fact perfect. If an induced subgraph H contains $\{a, b, e\}$, then $\omega(H) = 3 = \chi(H)$; otherwise one of $\{a, b, e\}$ will not be in H and so $\omega(H) \leq 2$ and without much difficulty we can show $\omega(H) = \chi(H)$. A few illustrative induced subgraphs are shown below.



$G_3[a, b, c, e]$



$G_3[a, c, d, e]$

$G_3[a, c, d]$

Theorem 6.14 A graph G is perfect if and only if $\overline{G}$ is perfect.

Theorem 6.15 A graph G is perfect if and only if no induced subgraph of G or $\overline{G}$ is an odd cycle of length at least 5.

Perfect Graphs

The following classes of graphs are known to be perfect:

- Trees
- Bipartite graphs
- Chordal graphs
- Interval graphs

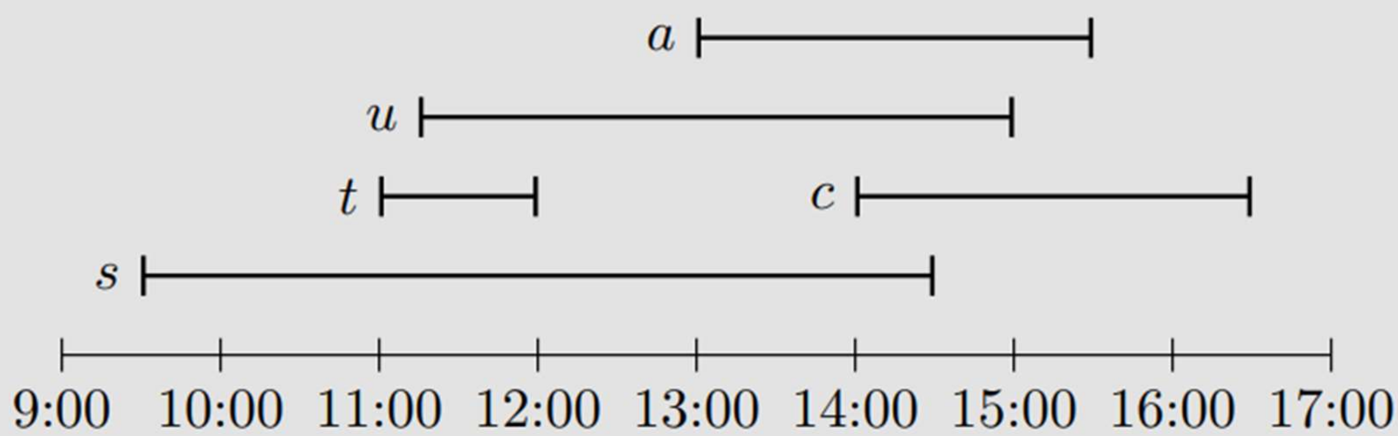
Definition 6.16 A graph G is ***chordal*** if any cycle of length four or larger has an edge (called a chord) between two nonconsecutive vertices of the cycle.

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 $I_x \cap I_y \neq \emptyset$.

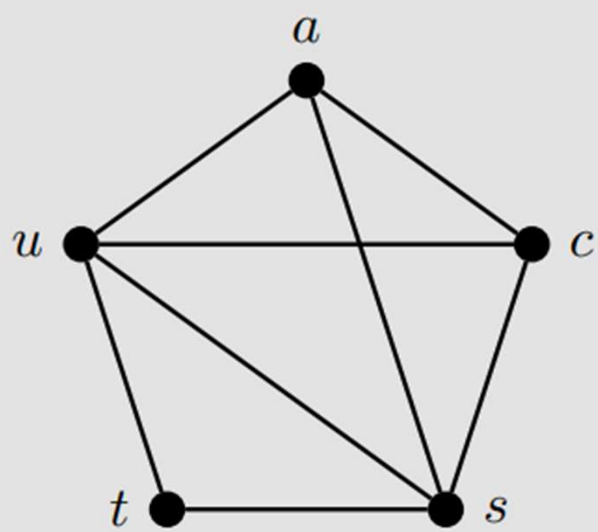
Example 6.7 Five student groups are meeting on Saturday, with varying time requirements. The staff at the Campus Center need to determine how to place the groups into rooms while using the fewest rooms possible. The times required for these groups is shown in the table below. Model this as a graph and determine the minimum number of rooms needed.

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

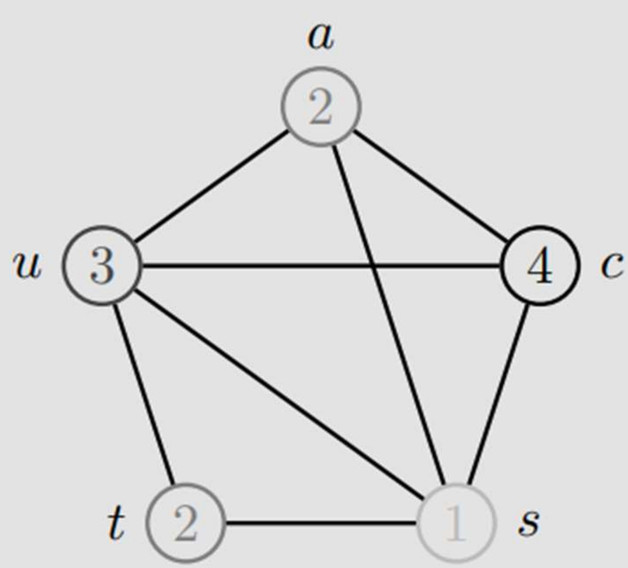
Solution: First we display the information in terms of the intervals. Although this step is not necessary, sometimes the visual aids in determining which vertices are adjacent.



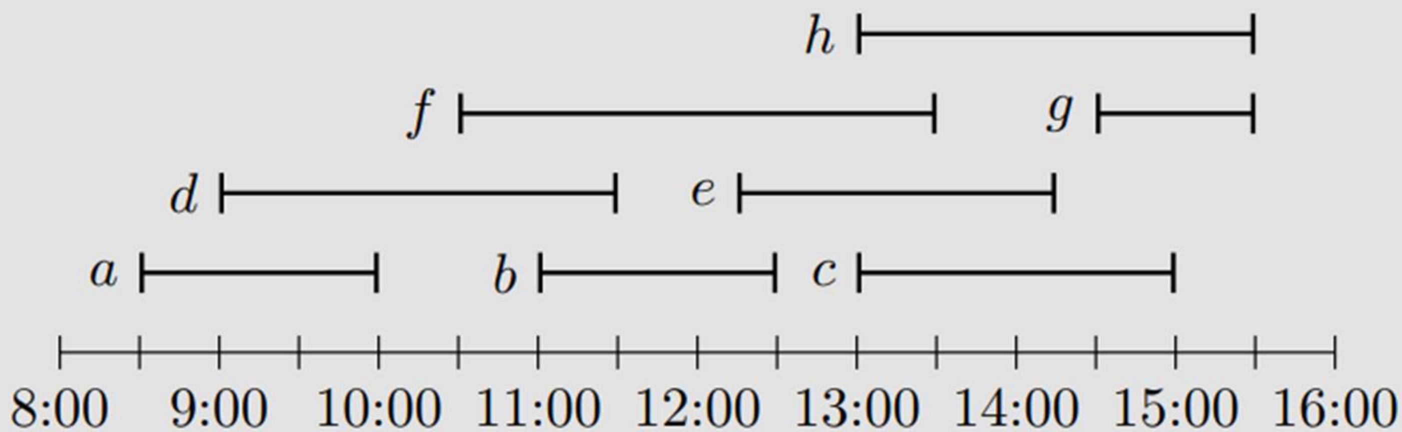
Below is the graph where each vertex represents a student group and two vertices are adjacent if their corresponding intervals overlap.



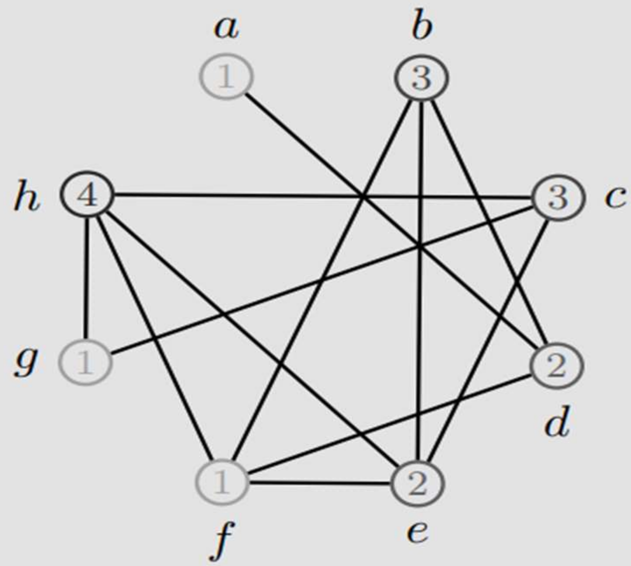
A proper coloring of this graph is shown below. Note that four colors are required since there is a K_4 subgraph with a, c, s, and u.



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.



Solution: Each meeting is represented by a vertex, with an edge between meetings that overlap and colors indicating the room in which a meeting will occur. If we color the vertices according to their start time (so in the order a, d, f, b, e, c, h, g), we get the coloring below.



Note that four meeting rooms are needed since there is a point at which four meetings are all in session, which is demonstrated by the K_4 among the vertices c, e, f, and h.

Lec # 38, 39 & 40

Edge Coloring

This section focuses on a different aspect of graph coloring where instead of assigning colors to the vertices of a graph, we will instead ***assign colors to the edges of a graph***. Such colorings are called edge-colorings and have their own set of definitions and notations, many of which are analogous to those for vertex colorings from above

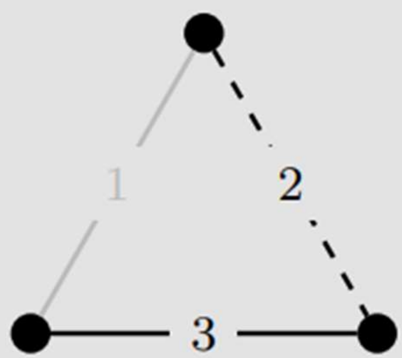
Definition 6.18 Given a graph $G = (V, E)$ an ***edge-coloring*** is an assignment of colors to the edges of G so that if two edges share an endpoint, then they are given different colors. The minimum number of colors needed over all possible edge-colorings is called the ***chromatic index*** and denoted $\chi'(G)$.

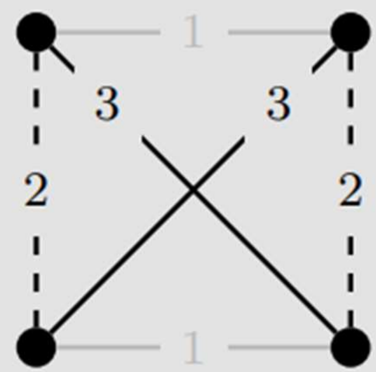
Example 6.9 Recall that the chromatic number for any complete graph is equal to the number of vertices. Find the chromatic index for K_n for all n up to 6.

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.

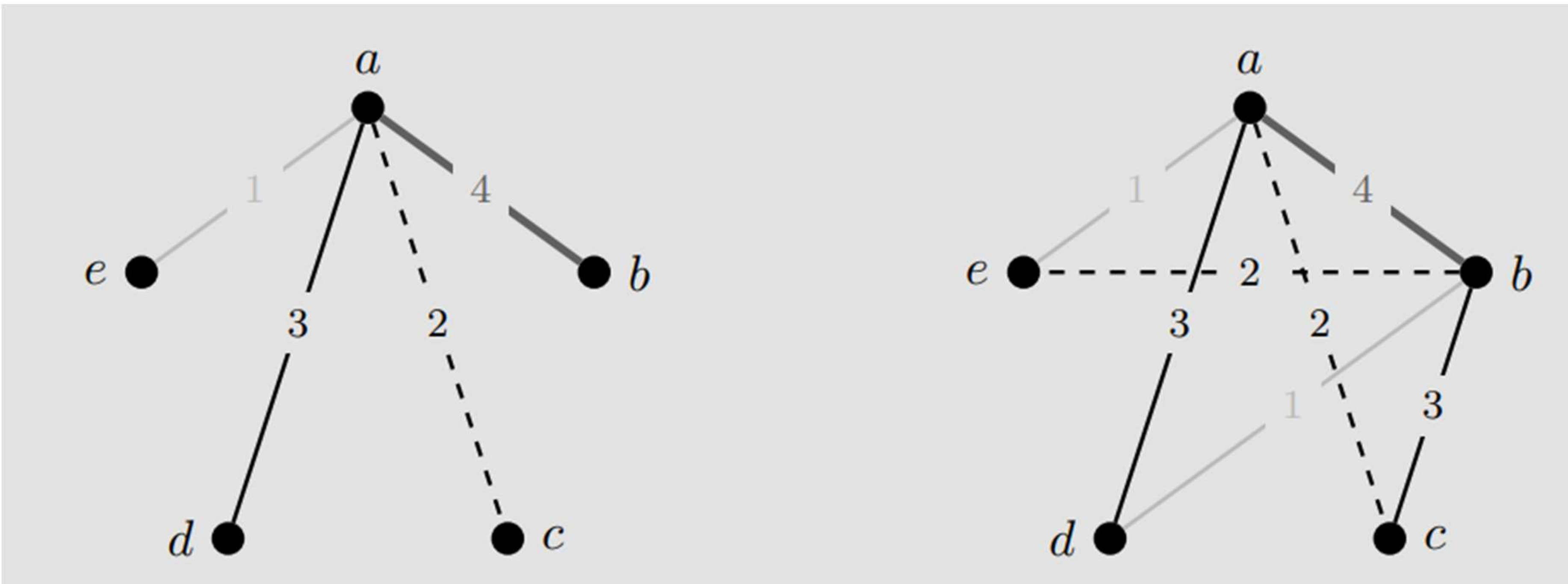
Solution: Since K_1 is a single vertex with no edges and K_2 consists of a single edge, we have $\chi'(K_1) = 0$ and $\chi'(K_2) = 1$. Due to their simplicity,

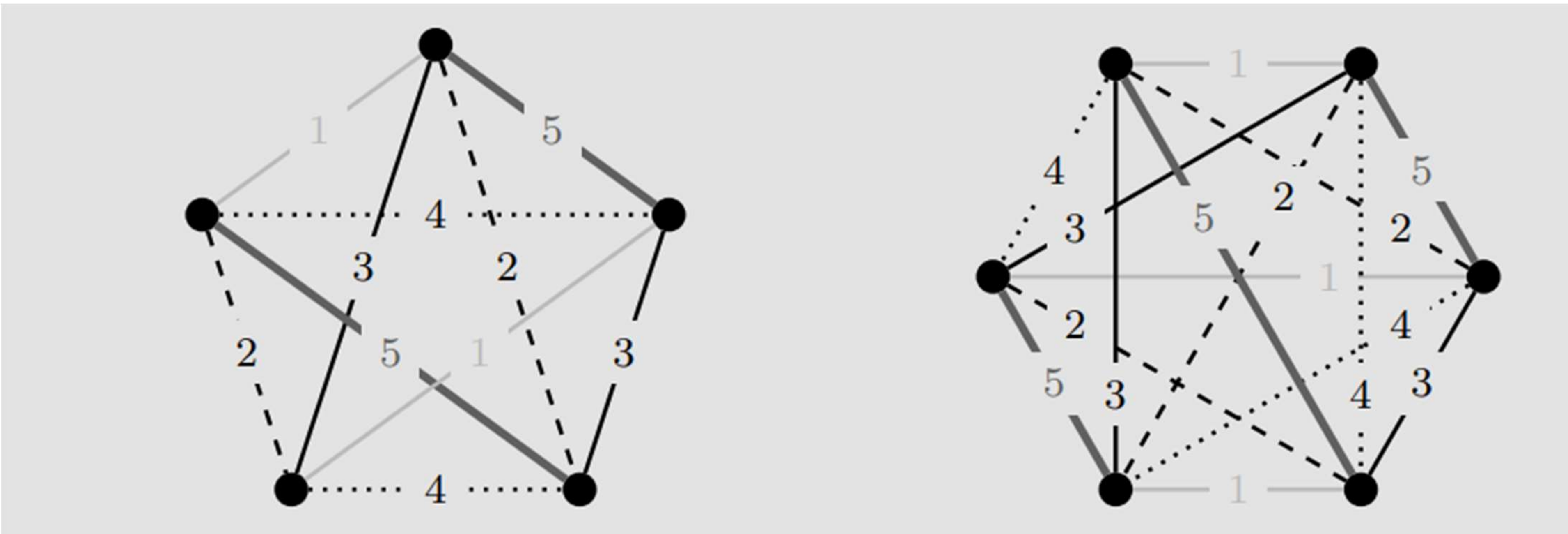
Example 6.9 Recall that the chromatic number for any complete graph is equal to the number of vertices. Find the chromatic index for K_n for all n up to 6.

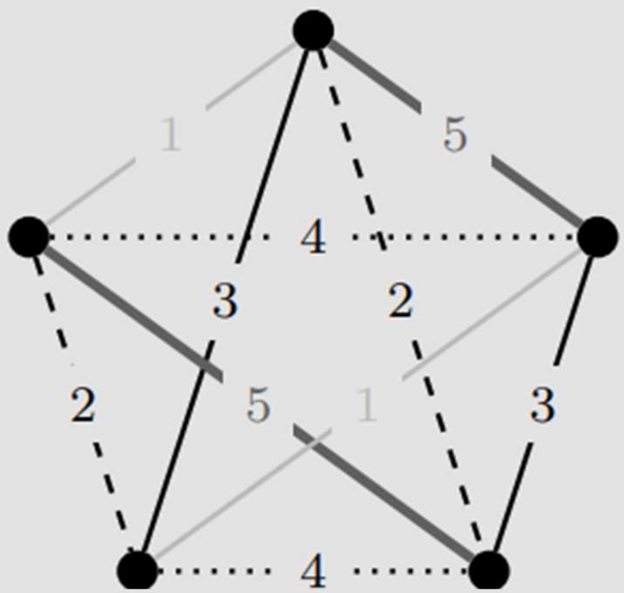


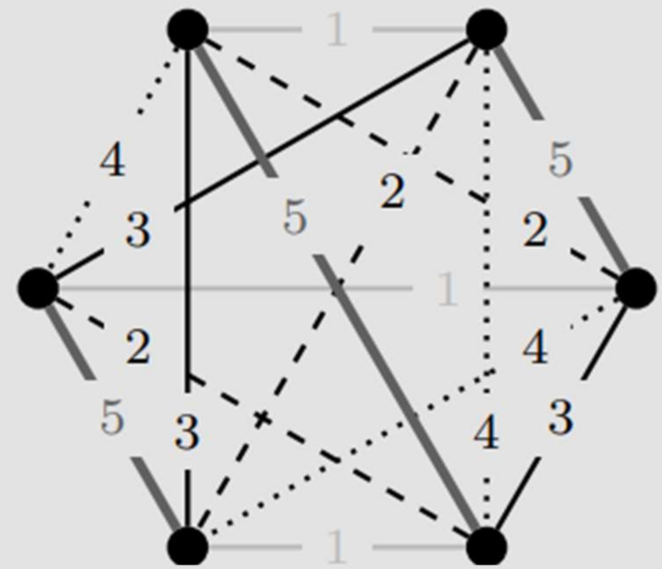


For K_3 since any two edges share an endpoint, we know each edge needs its own color and so $\chi'(K_3) = 3$. For K_4 we can color opposite edges with the same color, thus requiring only 3 colors. Optimal edge-colorings for K_3 and K_4 are shown below.


a
1
4
e
b
3
2
d
c
a
1
4
e
2
b
3
2
1
3
d
c


1
5
4
3
2
2
5
1
3
4
1
1
4
5
3
5
2
2
1
2
4
5
3
4
3
1

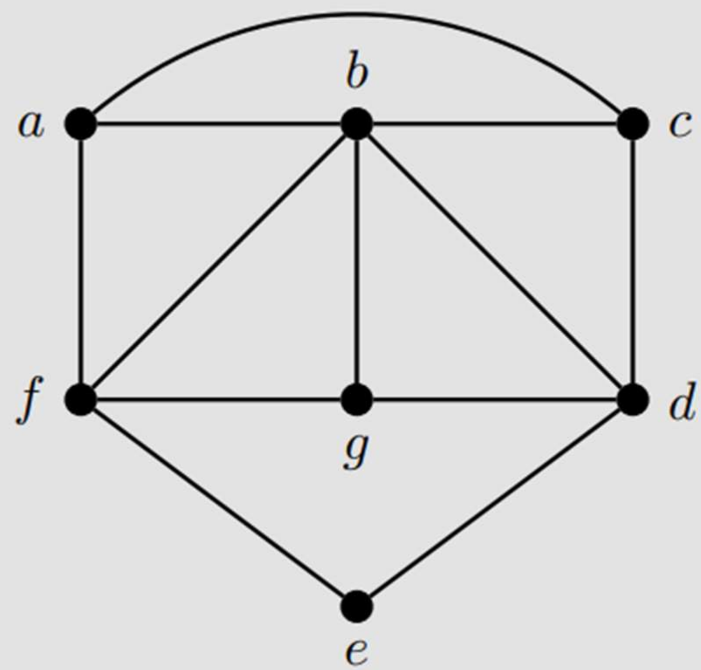




In general, $\chi'(K_n) = n - 1$ when n is even and $\chi'(K_n) = n$ when n is odd.

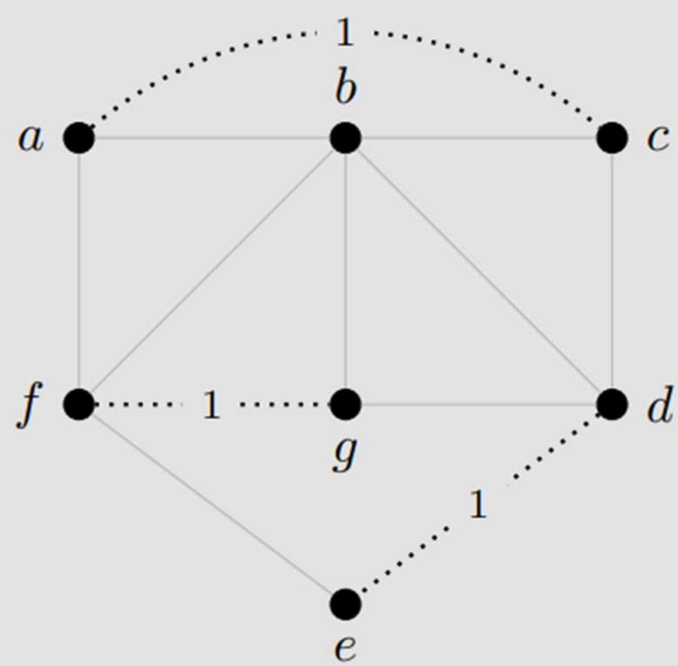
Theorem 6.19 (Vizing's Theorem) $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$ for all simple graphs G.

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.



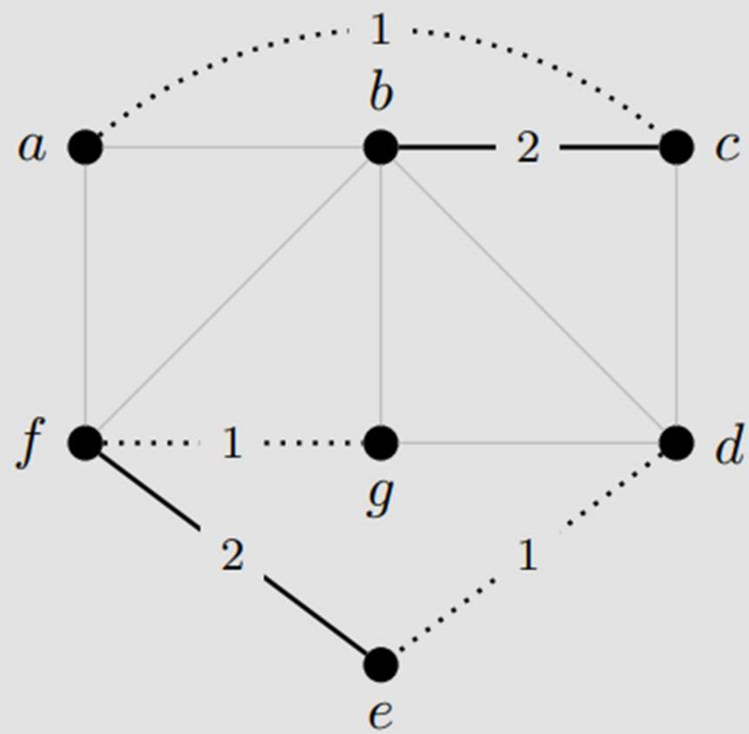
order $ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab$ using a greedy algorithm.

Step 1: Since the first three edges ac, fg, and de are not adjacent, we give each of them the first color.



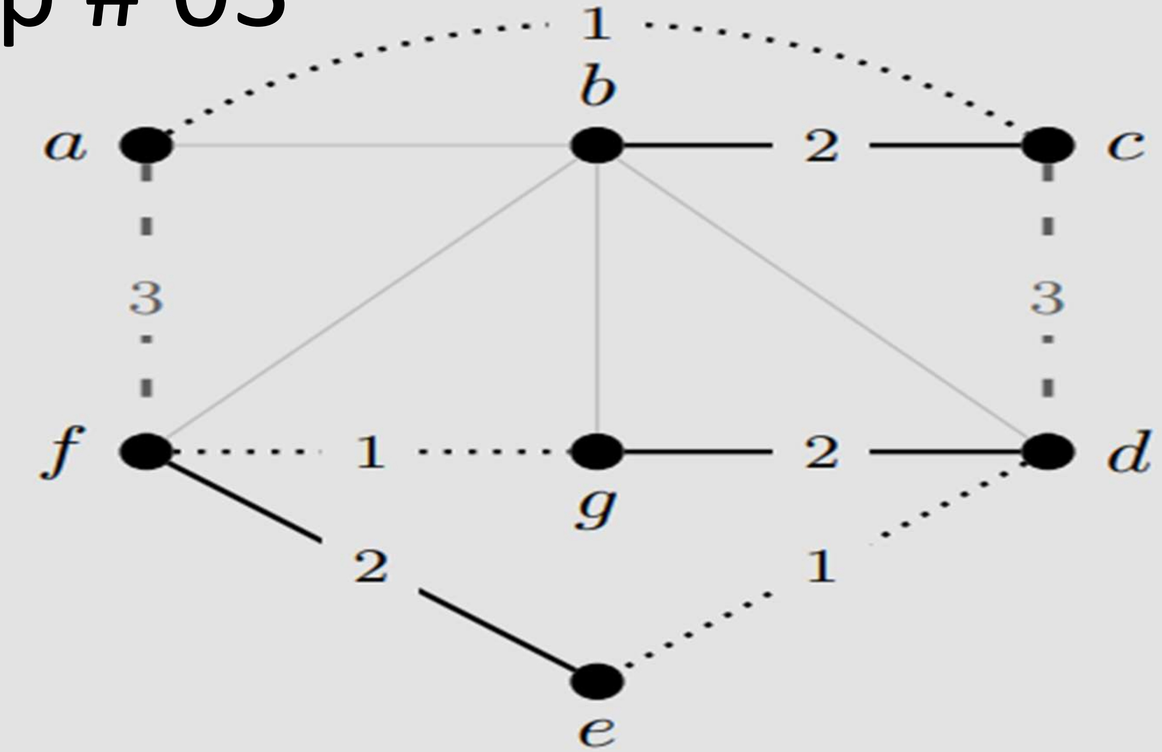
order $ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab$ using a greedy algorithm.

Step # 02



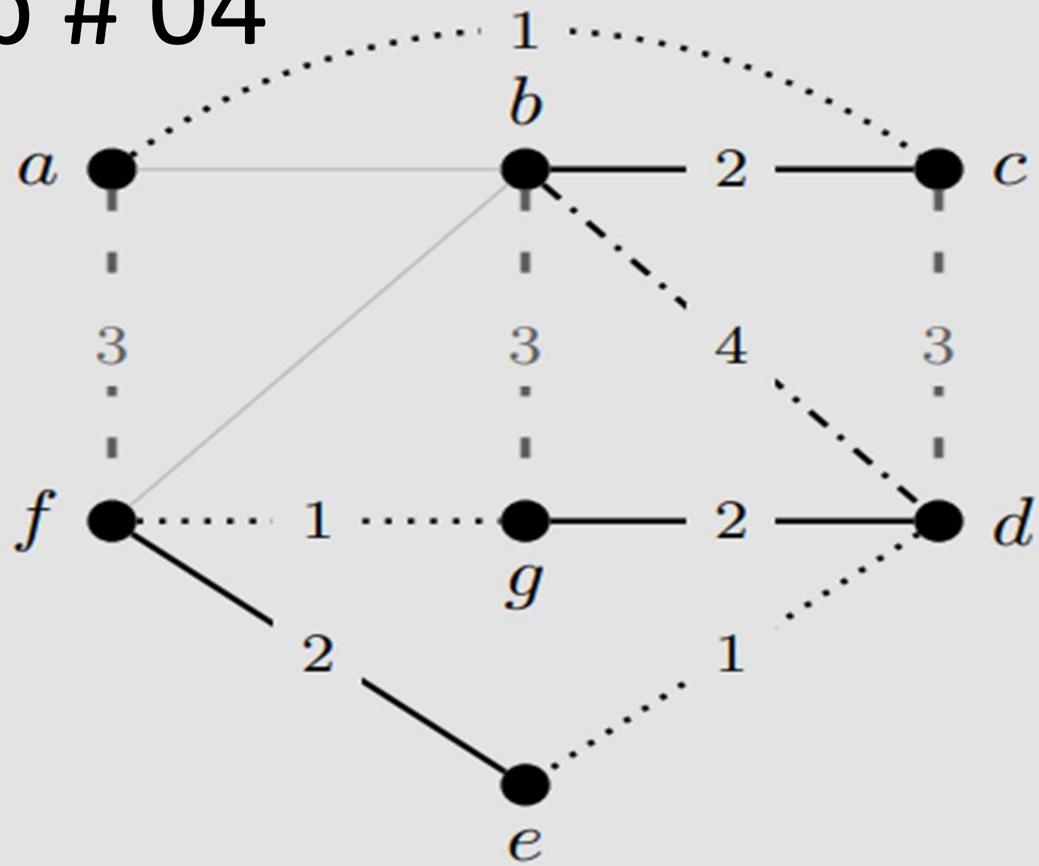
order $ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab$ using a greedy algorithm.

Step # 03



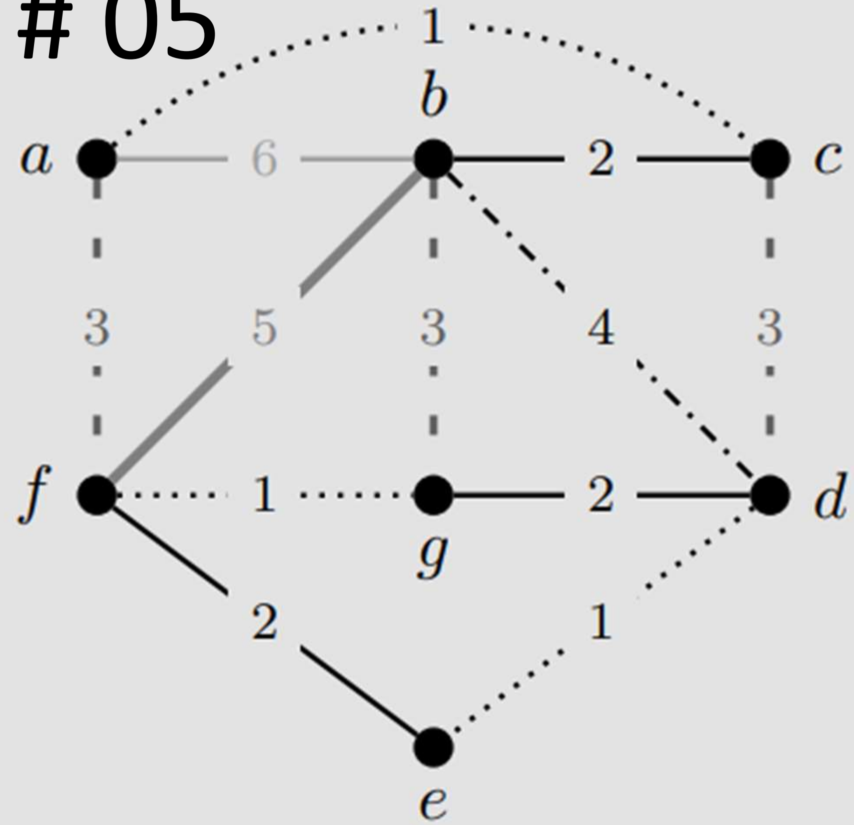
order $ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab$ using a greedy algorithm.

Step # 04

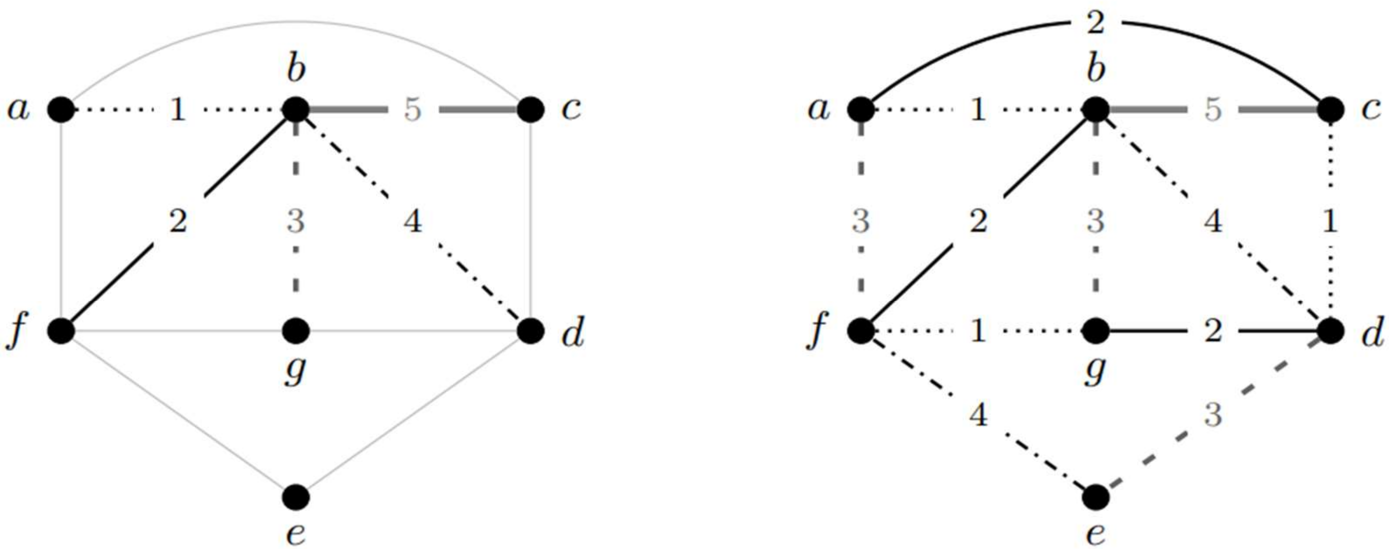


order $ac, fg, de, ef, bc, cd, dg, af, bd, bg, bf, ab$ using a greedy algorithm.

Step # 05



In the example above $\Delta(G_4) = 5$, and the edge-coloring above uses 6 colors; however $\chi'(G_4) = 5$. In general, starting with the vertex of highest degree and coloring its edges has a better chance of success in avoiding unnecessary colors, as shown below on the left. Once we have the minimum number of colors established, we attempt to fill in the remaining edges without introducing an extra color; one possible solution is shown below on the right.



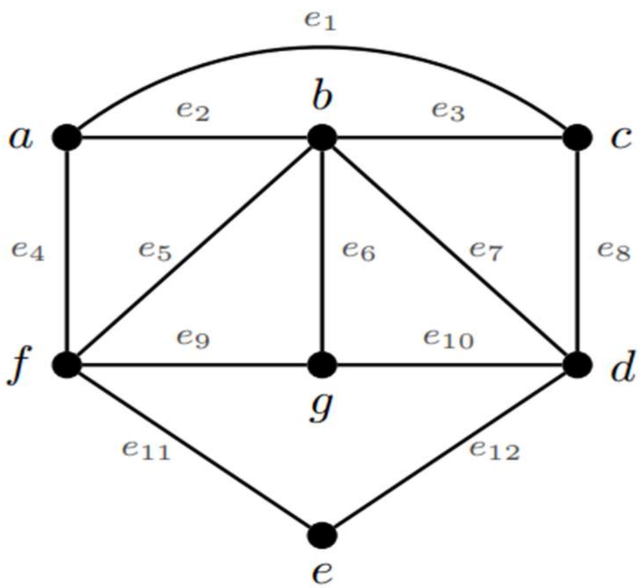
initial coloring of G_4 optimal edge-coloring of G_4

National University of computer and emerging sciences

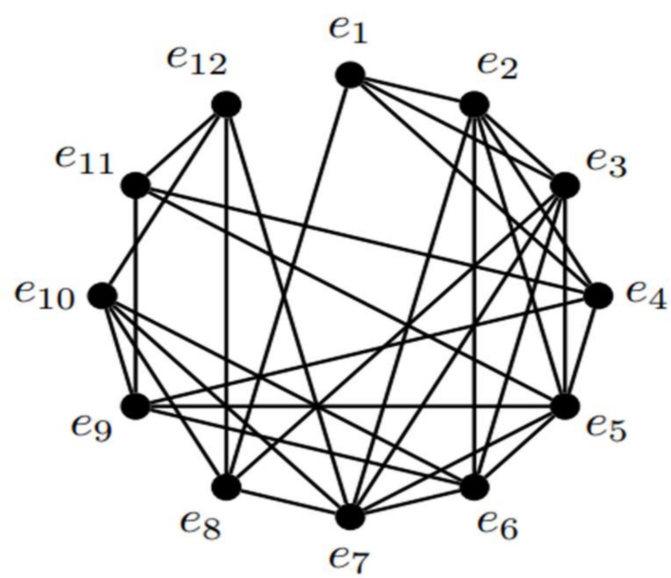
Foundation for Advancement of Science and Technology FAST

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.

Below is the graph G_4 from Example 6.10 and its line graph. Notice that the vertex e_1 in $L(G_4)$ is adjacent to e_2 and e_4 through the vertex a in G_4 and e_1 is adjacent to e_3 and e_8 through the vertex c in G_4.



G_4

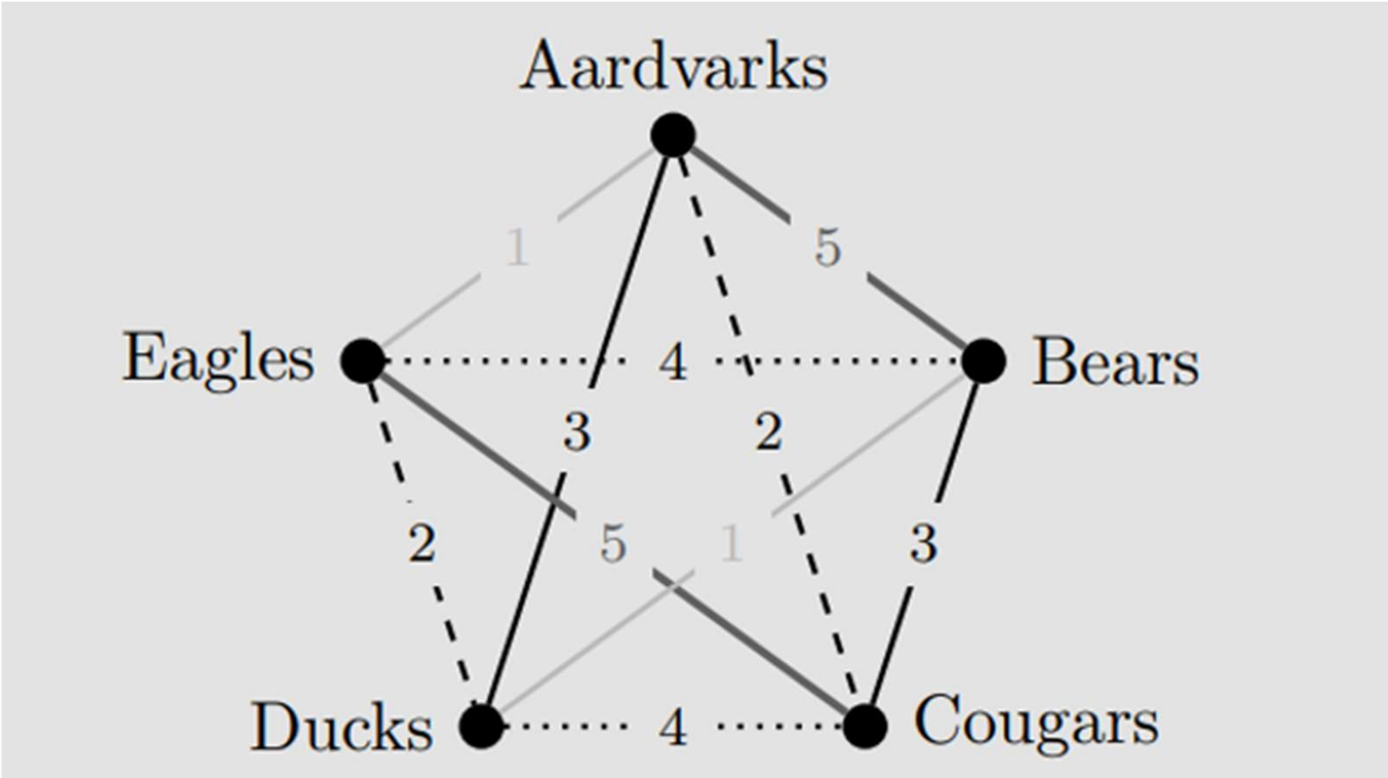


$L(G_4)$

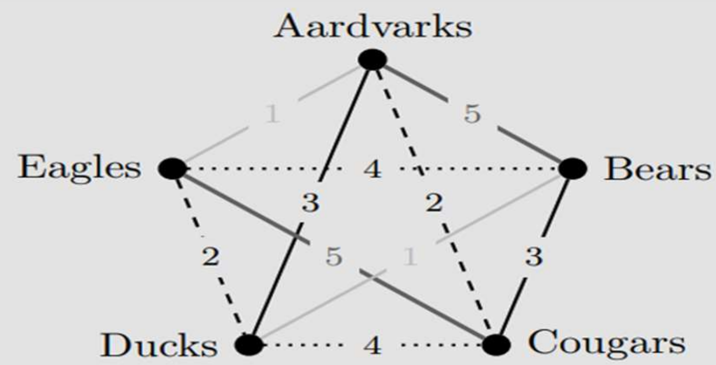
From this definition, it should be clear how edge-coloring and vertex-coloring are related. In particular, if edge e_1 is given the color blue in G, this would correspond to coloring vertex e_1 in $L(G)$ with blue. This correspondence provides the following result.

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 (Aardvarks, Bears, Cougars, Ducks, and Eagles) 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.



In general, $\chi'(K_n) = n - 1$ when n is even and $\chi'(K_n) = n$ when n is odd.



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

Coloring Variations

This section will highlight additional variations of graph coloring, specifically vertex colorings. Within each of these we will see applications of graph coloring that can inform why this new version of coloring is worthy of study.

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 $G[x_1, \ldots, x_i]$ which consists of the vertices up to and including x_i. The maximum number of colors a specific algorithm $\mathcal{A}$ uses on any possible ordering of the vertices is denoted $\chi_{\mathcal{A}}(G)$.

A **greedy** algorithm called ***First-Fit*** that uses the first available color for a new vertex

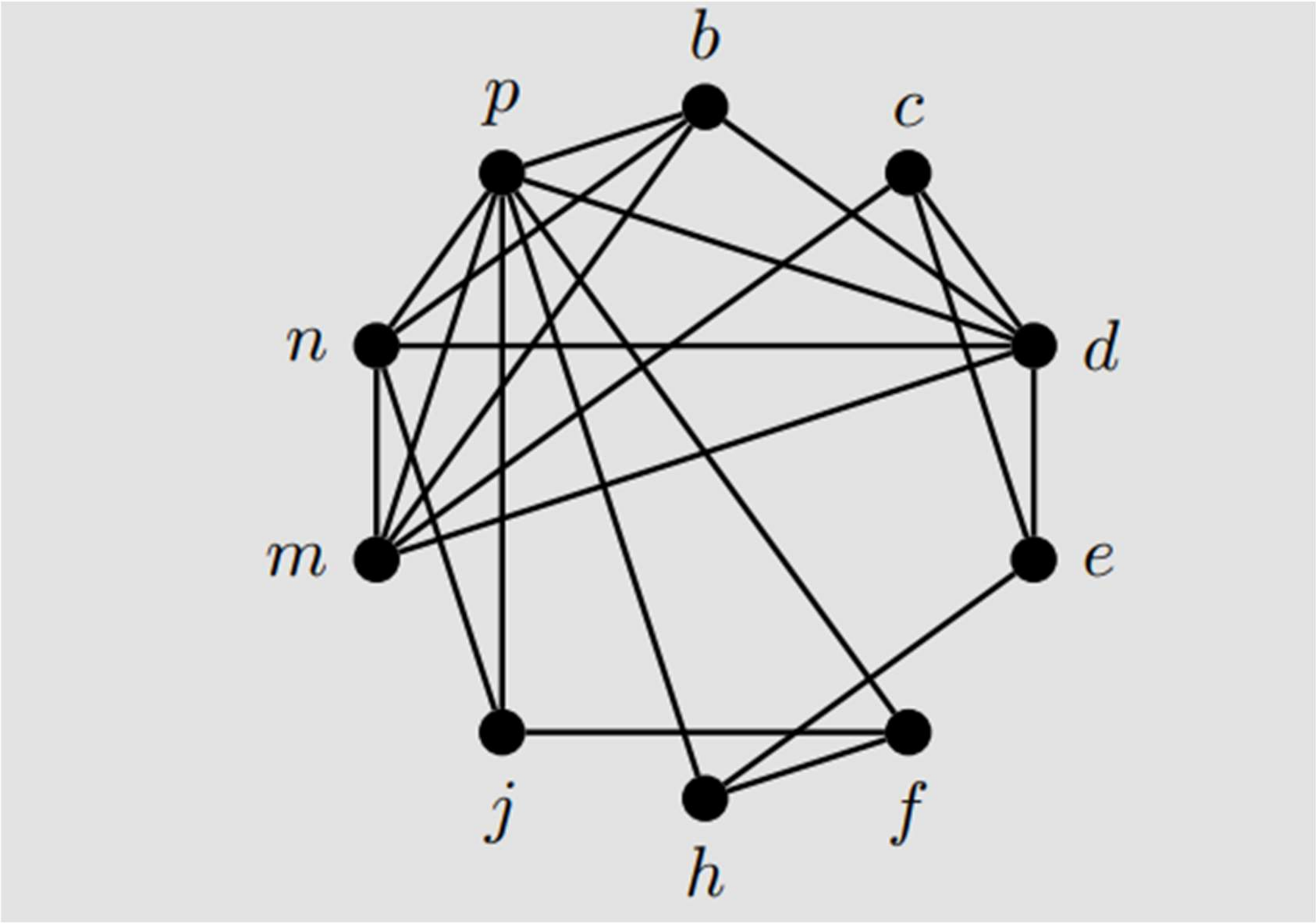
First-Fit Coloring Algorithm

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.


b
p
c
n
d
m
e
j
h
f

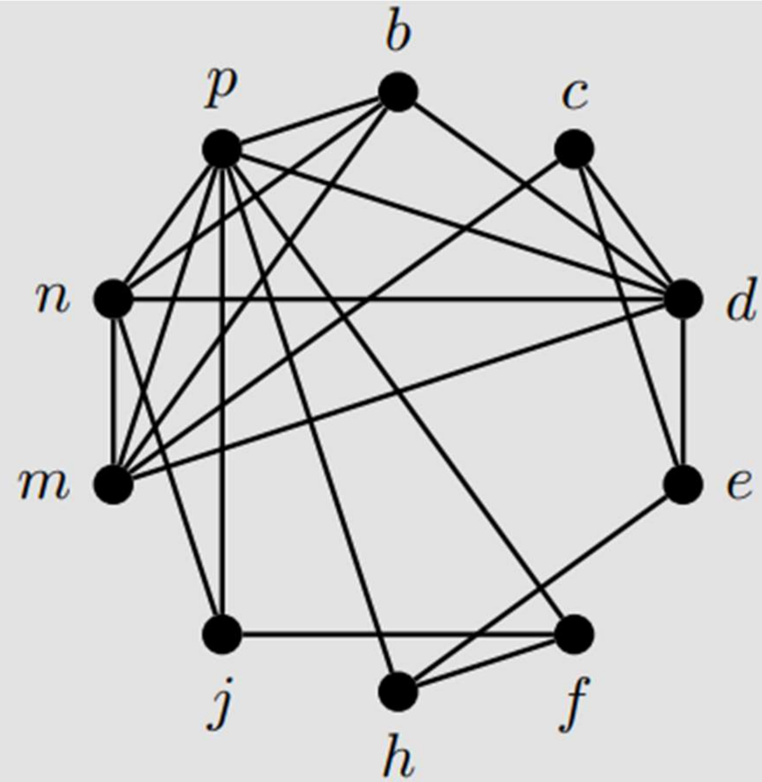
First-Fit Coloring Algorithm

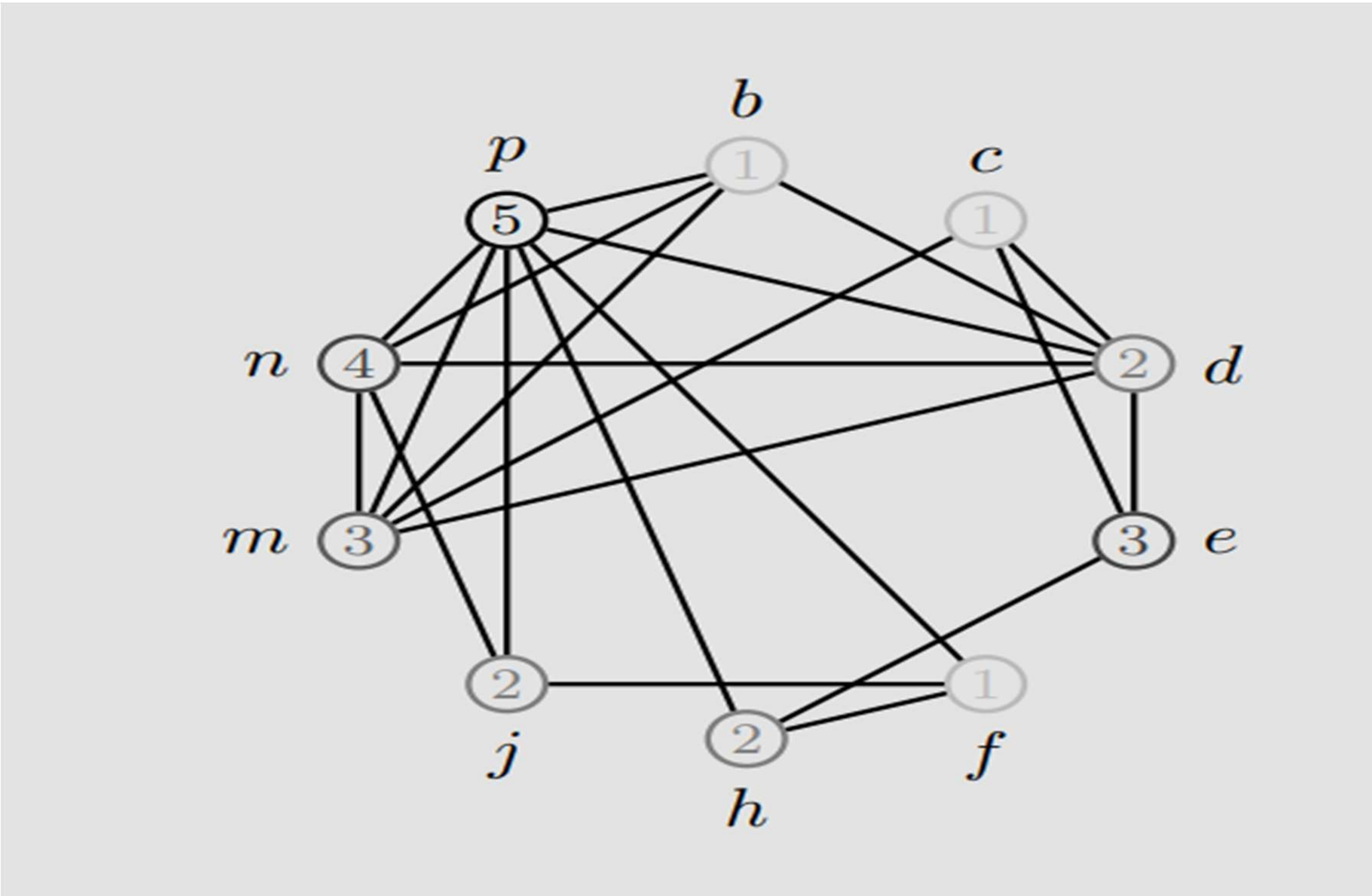
Input: Graph G with vertices ordered as $x_1 \prec x_2 \prec \cdots \prec x_n$.

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.

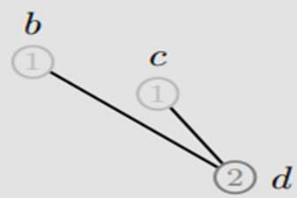



b
p
c
1
5
1
n
4
2
d
m
3
3
e
2
1
2
j
f
h

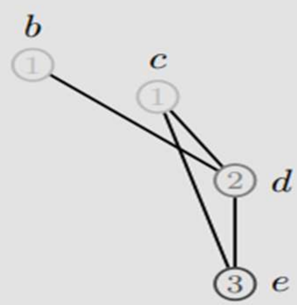
Hint:

Step 2: Color c with 1 since b and c are not adjacent.

Step 3: Color d with 2 since d is adjacent to a vertex of color 1.



Step 4: Color e with 3 since e is adjacent to a vertex of color 1 (c) and a vertex of color 2 (d).



Do Example 6.15 & 6.16