Chapter 6: Colorings

6.1 Map Colorings

Given a map with several countries, we may ask how many colors are required to paint the map so that no two countries with a boundary in common are assigned the same color. Historically, there was a conjecture called "four-color problem", that is, four colors are sufficient to paint any map. We start the story with some formal definitions.

Definition 6.1.1: A *map* is defined to be a plane connected graph with no bridges. A map G is said to be *k-face colorable* if we can color its faces with at most k colors in such a way that no two *adjacent faces*, which share a common edge, have the same color.

Four Color Conjecture: *Every map is 4-face colorable.*

Given a map G, its dual G^* is also a connected plane graph. In order to paint the faces (countries) of G, it is equivalent to paint the vertices of G^* such that no adjacent vertices are assigned the same color. Thus, the map coloring problem is equivalent to vertex coloring problem of plane graphs.

6.2 Vertex Colorings

Definition 6.2.1: Let $G = (V, E, \phi)$ be a graph. A *(proper) vertex coloring* of G is a mapping $c : V \to \mathbb{N}$ that satisfies

$$c(u) = c(v) \Rightarrow \{u, v\} \notin \phi(E) \ \forall u, v \in V;$$

that is, if u and v are adjacent, then they receive different colors. When there is no ambiguity, the word *coloring* would mean vertex coloring.

For $k \in \mathbb{N}$, a *(proper) k-coloring* of G is a coloring $c : V \to \{1, \dots, k\}$. We say that G is *k-vertex colorable* or *k-colorable* if G has a (proper) k-coloring.

Clearly, a simple graph of order p is p-colorable.

Definition 6.2.2: Let G be a simple graph. The least integer k such that G is k-colorable is called the *chromatic number* of G and is denoted by $\chi(G)$.

Suppose G has a k-coloring and let V_i be the set of vertices colored by i. Thus, there exists a partition $(V_1, V_2, \dots, V_k)$ of V, where V_i may be empty, such that each V_i is an *independent set* (i.e., any two distinct vertices in V_i are not adjacent). We also call the partition $(V_1, V_2, \dots, V_k)$ a k-coloring of G. Moreover, $\chi(G) = k$ implies that each subset V_i of a k-coloring $(V_1, V_2, \dots, V_k)$ is not empty.

It is obvious that if the simple graph G has ω components $G_1, \dots, G_\omega$, then

$$\chi(G) = \max\{\chi(G_i) \mid 1 \le i \le \omega\}.$$

Therefore, we assume all the graphs being considered in coloring problem are connected.

Example 6.2.3: By the definition of chromatic number of graph, we have

1. $\chi(G) = 1$ if and only if $G \cong N_p$ for some $p \in \mathbb{N}$.

2. $\chi(K_p) = p$ for $p \in \mathbb{N}$.

3. $\chi(P_p) = 2$ for $p \geq 2$.

4. For $p \geq 3$, $\chi(C_p) = \begin{cases} 2 & \text{if } p \text{ is even,} \\ 3 & \text{if } p \text{ is odd.} \end{cases}$

Definition 6.2.4: A graph G is called *critical* if $\chi(H) < \chi(G)$ for every proper subgraph H of G. A critical graph with $\chi(G) = k$ is called a k-*critical* graph.

From the definition above, every critical graph is connected.

Lemma 6.2.5: *Every graph with chromatic number k has a k-critical subgraph.*

Lemma 6.2.6: *If G is k-critical, then $\delta(G) \geq k - 1$.*

Theorem 6.2.7: *For any simple graph G with maximum degree Δ, we have $\chi(G) \leq \Delta + 1$.*

Brook's Theorem (1941): *For a connected simple graph G that is neither a complete graph nor an odd cycle, we have $\chi(G) \leq \Delta(G)$.*

The following are some important results about coloring problem of plane graphs.

Theorem 6.2.8:

(a) A map G is k-face colorable if and only if its dual G^* is k-vertex colorable.

(b) Let G be a connected plane graph without loops. Then G has a k-vertex coloring if and only if its dual G^* has a k-face coloring.

Theorem 6.2.9: *Every plane graph is 6-colorable.*

A *Jordan curve* is a continuous non-self-intersection closed curve. Let J be a Jordan curve in the plane. It partitions the plane into two disjoint open sets, the *interior* and *exterior* of J, which are denoted by $\text{int}(J)$ and $\text{ext}(J)$ respectively.

Jordan Curve Theorem: *Let J be a Jordan curve. Any curve joining a point in int(J) to a point in ext(J) must meet J at some point.*

Theorem 6.2.10: *Every plane graph is 5-colorable.*

6.3 Algorithms on Vertex Coloring

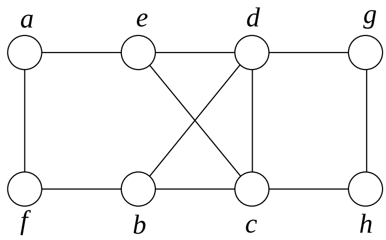
Algorithm 6.3.1 (Sequential Coloring)**:**

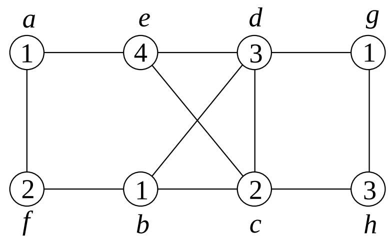
Step 1. List the vertices of G as $u_1, \ldots, u_p$. List the colors available as $1, \ldots, p$.

Step 2. Color the first vertex by 1.

Step 3. Color each subsequent vertex by the color of smallest number that is not used by any of its neighbors. Repeat this until all vertices have been colored.

Example 6.3.1: Consider the graph below. We use the alphabetical ordering a, b, c, d, e, f, g, h as the list. Apply the sequential coloring, vertex a is colored by 1 and then vertex b is colored by 1, because b is not a neighbor of a. Next we color c by 2 and so on. Finally we obtain a 4-coloring of the graph and it is labelled as the graph on right hand side.





As a result, we have $\chi \leq 4$.

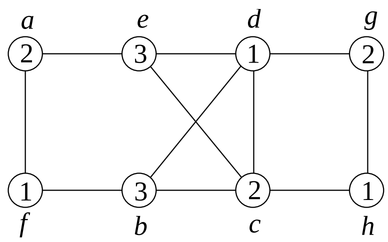
The *largest-first sequential coloring* is an algorithm that only differ from the sequential coloring in the first step.

Algorithm 6.3.2 (Largest-First Sequential Coloring, Welsh and Powell)**:**

Step 1. List the vertices of G as $u_1, \ldots, u_p$ so that $\deg(u_1) \geq \cdots \geq \deg(u_p)$. In practice, we always have a pre-ordering (priority) for vertices.

Continue with Steps 2 and 3 of the sequential coloring.

Example 6.3.2: Consider the graph in Example 6.3.1. We use the reverse alphabetical ordering h, g, f, e, d, c, b, a as the pre-ordering of the vertices. From Step 1 we obtain the ordering d, c, e, b, h, g, f, a. So, we first color d by 1 and then color c by 2, e by 3, etc. Finally we have the following 3-coloring.



Hence we have $\chi \leq 3$. Since the graph is not bipartite, we have $\chi = 3$.

Theorem 6.3.3: *If a graph G has a degree sequence $(d_1, \ldots, d_p)$, then there is a greedy coloring of G that uses at most $\max\limits_{1\leq i\leq p}\big\{\min\{i, d_i+1\}\big\}$ colors, that is, $\chi(G) \leq \max\limits_{1\leq i\leq p}\big\{\min\{i, d_i+1\}\big\}$.*

Proof: According to Algorithm 6.3.2, the color assigned to vertex u_i (with degree d_i) is the minimum of i and $d_i + 1$. Therefore, the result follows by taking maximum of these numbers. $\square$

The smallest-first sequential coloring is also modified from the sequential coloring. We create a list by first choosing the vertex of minimum degree in G as u_p. Then we choose a vertex of minimum degree in $G - \{u_p\}$ as u_{p-1}, which is the second last in the list. Continue in this way until all vertices are chosen.

Algorithm 6.3.3: (Smallest-Last Sequential Coloring)

Step 1. (a) Choose a vertex of minimum degree in G as u_p.

(b) For $i = p-1, p-2, \ldots, 1$, choose a vertex of minimum degree in $G - \{u_p, \ldots, u_{i+1}\}$ as u_i.

(c) List the vertices as $u_1, \ldots, u_p$.

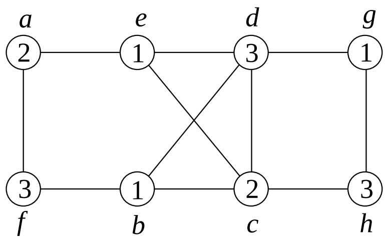
(d) List the colors available as $1, \ldots, p$.

Continue with Steps 2 and 3 of the sequential coloring.

Example 6.3.4: Consider the graph in Example 6.3.1. We pre-order the vertices by alphabetical ordering a, b, c, d, e, f, g, h. From Step 1(a) - (d) we obtain

$$u_8 = h, u_7 = g, u_6 = f, u_5 = a, u_4 = e, u_3 = d, u_2 = c, u_1 = b.$$

Therefore, we have the following 3-coloring.



Given a partial coloring, the *color-degree* of a vertex v is the number of distinct colors that have been assigned to the neighbors of v.

Algorithm 6.3.4: (Maximum Color-Degree Coloring)

Step 1. Create a list U that contains vertices in descending order of degree.

Step 2. Color the first vertex v by 1 and then delete v from U.

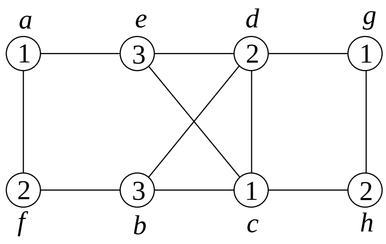
Step 3. Choose a vertex $w \in U$ with maximum color-degree. When there is a tie, choose the vertex that appears earliest on U. Color w with the smallest color that have not been used by any of its neighbors. Delete w from U and repeat Step 3 until $U = \varnothing$.

Example 6.3.5: Consider the graph in Example 6.3.1. Suppose we pre-ordered the vertices in alphabetical order. Then we obtain the list $U = (c, d, b, e, a, f, g, h)$. The following table shows the procedures of this coloring.

Sequence with respect to color-degree	Value of the coloring c	Remark
c, d, b, e, a, f, g, h	$c(c) = 1$	
$d(1), b(1), e(1), h(1), a, f, g$	$c(d) = 2$	d, b, e and h are of color-degree 1, but d is in the highest piority
$b(1, 2), e(1, 2), g(2), h(1), a, f$	$c(b) = 3$	b and e are of color-degree 2, but b is in higher piority
$e(1, 2), f(3), g(2), h(1), a$	$c(e) = 3$	
$a(3), f(3), g(2), h(1)$	$c(a) = 1$	a, f, g and h are tie
$f(1, 3), g(2), h(1)$	$c(f) = 2$	
$g(2), h(1)$	$c(g) = 1$	g and h are tie
$h(1)$	$c(h) = 2$	

Remark: Numbers inside the parentheses are the colors assigned to the neighbors of that vertex.

Hence we have the following coloring.



The *independent number* of a graph G, denoted by $\alpha(G)$, is the maximum size of an independent set. An independent set of G of size $\alpha(G)$ is called a *maximum independent set* of G.

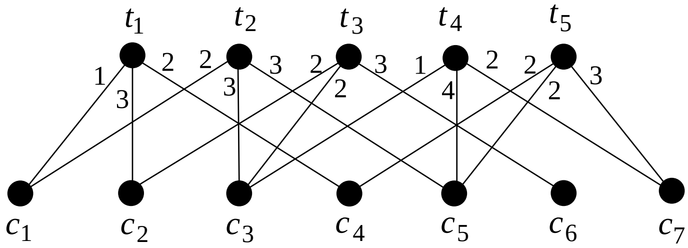
Proposition 6.3.6: *The set S is an independent set of G if and only if each edge has at least one end in $V(G) \setminus S$.*

Lemma 6.3.7: *If G is a graph of order p, then $\chi(G) \geq \frac{p}{\alpha(G)}$.*

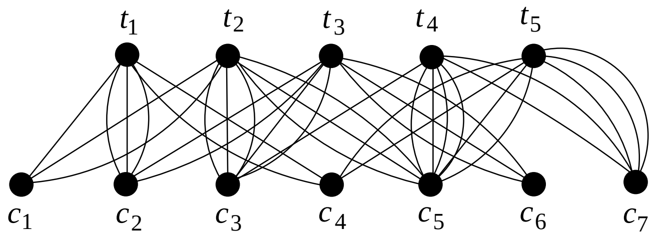
Corollary 6.3.8: *If G is a graph of order p, then $\chi(G) \leq p - \alpha(G) + 1$.*

6.4 Edge Colorings

Example 6.4.1: Suppose there is a mini-school that have 5 teachers and 7 classes. These five teachers t_1, t_2, t_3, t_4, t_5 meet with classes $c_1, \ldots, c_7$ every week. Suppose teacher t_i must meet with class c_j a total number of w_{ij} periods (for example, one period is a one-hour lesson) per week. To keep track of possible teachers for each class, both t_i $(1 \leq i \leq 5)$ and c_j $(1 \leq j \leq 7)$ are represented as vertices. Note that all edges connect a teacher to a class, but not teacher-to-teacher or class-to-class. So the graph is bipartite. We put the weight w_{ij} on the edge $t_i c_j$. For example, we have the following weighted graph.



Actually, we may redraw the above graph as a multigraph below, where each edge represents a one-hour lesson. For example, there is only one edge incident with c_1 and t_1; there are three edges incident with c_2 and t_1, etc.



Our problem is to determine the minimum number of periods per week that must appear in the timetabling grid.

Note that a teacher cannot meet two classes at the same period. Also, two teachers may not meet the same class at the same time. If we represent a period by a color, then the problem transformed into assigning colors to edges so that two edges with a common vertex will receive different colors. Our goal becomes finding the minimum number of colors satisfying the above requirement.

Recall that, two edges are said to be adjacent if they have a common end vertex.

Definition 6.4.2: Let $G = (V, E, \phi)$ be a graph. A *(proper) edge coloring* of G is a mapping $c : E \to \mathbb{N}$ satisfying

$$c(e) = c(f) \Rightarrow \phi(e) \cap \phi(f) = \varnothing \ \forall e, f \in E;$$

i.e., no adjacent edges receive the same color.

For $k \in \mathbb{N}$, a *(proper) k-edge coloring* of G is an edge-coloring $c : E \to \{1, \ldots, k\}$. We say that G is *k-edge colorable* if G has a (proper) k-edge coloring.

Definition 6.4.3: For a graph G, the least number $k \in \mathbb{N}$ such that G is k-edge colorable is called the *edge chromatic number* of G, or the *chromatic index* of G, and is denoted by $\chi'(G)$.

By the definition of edge-coloring we have the following observation.

Proposition 6.4.4: *For any graph G, $\chi'(G) \geq \Delta(G)$.*

Example 6.4.5: For $p \in \mathbb{N}$, $\chi'(K_p) = \begin{cases} p & \text{if } p \text{ is odd,} \\ p-1 & \text{if } p \text{ is even.} \end{cases}$

Theorem 6.4.6: *For a bipartite graph G, we have $\chi'(G) = \Delta(G)$.*

Corollary 6.4.7: $\chi'(K_{m,n}) = \max\{m, n\}$.

Theorem 6.4.8 (Vizing)**:** *If G is a simple graph, then $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$.*

Go back to Example 6.4.1, the maximum degree $\Delta = 9$ occurs at c_5. Theorem 6.4.6 implies that 9 periods is enough. The follow-up question is how to design the timetable with 9 time slots. We introduce an algorithm that was provided by W.H. Chan and W.C. Shiu*.

Chan-Shiu Algorithm: Suppose G is a bipartite multigraph with bipartition (T, C). Suppose the maximum degree of G is Δ and it occurs at a vertex in C.

Step 1. Assign Δ numbers (colors) $1, 2, \ldots, \Delta$ to G in which no two edges of same number are incident to a common vertex in C.

Step 2. If there is no vertex in T that is incident to two or more edges of same number, then stop and we obtain a required assignment. Otherwise, let $t_i \in T$ that is incident with two or more edges of same number. Let this number be n_1.

Step 3. Choose a number n_2 that has not been assigned to the edges being incident with t_i.

Step 4. Find the component that contains t_i and formed by the edges assigned by n_1 or n_2. [By breadth-first search, starting from t_i, along the edges assigned by n_1 or n_2.]

Step 5. If the component is Eulerian, then go to Step 6. Otherwise, add an artificial vertex and edges to join the artificial vertex and the vertex of odd degree.

Step 6. Find the Eulerian tour of the component by Fleury's algorithm.

Step 7. Reassign the two numbers alternately to the edges of the component along the Eulerian tour from the artificial vertex. If no artificial vertex was added, start from the vertex of maximum degree.

Step 8. Remove the artificial vertex and go to Step 2.

*Wai Hong Chan, Solving Time-tabling Problem by Graph Theory, *Thesis of Bachelor of Science in Combined Science, Mathematical Science Major*, Hong Kong Baptist College, 1994, supervised by Wai Chee Shiu.

Example 6.4.9: Apply Step 1 of Chan-Shiu Algorithm to the graph in Example 6.4.1, numbers $1, 2, 3$ were assigned to the edges incident to c_1 and numbers $4, 5, 6, 7, 8$ to the edges incident to c_2 and so on. Precisely, we list all edges and their corresponding number (in parentheses) as follows:

$$\begin{array}{ccccccccc} c_1t_1(1) & c_1t_2(2) & c_1t_2(3) & c_2t_1(4) & c_2t_1(5) & c_2t_1(6) & c_2t_3(7) & c_2t_3(8) & c_3t_2(9) \\ c_3t_2(1) & c_3t_2(2) & c_3t_3(3) & c_3t_3(4) & c_3t_4(5) & c_4t_1(6) & c_4t_1(7) & c_4t_5(8) & c_4t_5(9) \\ c_5t_2(1) & c_5t_2(2) & c_5t_2(3) & c_5t_4(4) & c_5t_4(5) & c_5t_4(6) & c_5t_4(7) & c_5t_5(8) & c_5t_5(9) \\ c_6t_3(1) & c_6t_3(2) & c_6t_3(3) & c_7t_4(4) & c_7t_4(5) & c_7t_5(6) & c_7t_5(7) & c_7t_5(8) & \end{array}$$

Besides, we present the assignment (coloring) as follows:

	t_1	t_2	t_3	t_4	t_5
c_1	1	$2, 3$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$9, 1, 2$	$3, 4$	5	
c_4	$6, 7$				$8, 9$
c_5		$1, 2, 3$		$4, 5, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$6, 7, 8$

From Steps 2 and 3, we choose t_2 which is incident to 3 edges that were assigned 2. Also, number 4 has not been assigned to edges incident with t_2.

Performing Step 4, we have $t_2c_1(2), t_2c_3(2), t_2c_5(2)$, $c_3t_3(4), c_5t_4(4)$, $t_3c_6(2)$ and $t_4c_7(4)$. It is not Eulerian and we add an artificial vertex a and edges ac_1, ac_6, ac_7, at_2 to make this component to be Eulerian with an Eulerian tour $ac_1t_2ac_6t_3c_3t_2c_5t_4c_7a$.

Performing Step 7, we have $ac_1(2), c_1t_2(4), t_2a(2)$ and so on. Then we have

	t_1	t_2	t_3	t_4	t_5
c_1	1	$\boxed{4}, 3$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$3, 4$	5	
c_4	$6, 7$				$8, 9$
c_5		$1, \boxed{4}, 3$		$\boxed{2}, 5, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$6, 7, 8$

Boxed numbers have been changed.

Go back to Step 2, suppose we choose t_1 with assigned number $n_1 = 6$ and choose another number $n_2 = 9$. Performing Step 4, we have a component induced by $t_1c_2(6)$, $t_1c_4(6)$, $c_4t_5(9)$, $t_5c_5(9)$, $t_5c_7(6)$ and $c_5t_4(6)$. After adding an artificial vertex a, we have an Eulerian tour $at_5c_5t_4ac_7t_5c_4t_1c_2a$.

Performing Step 7, start from a and follow the Eulerian tour above, we have

	t_1	t_2	t_3	t_4	t_5
c_1	1	$3, 4$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$3, 4$	5	
c_4	$\boxed{9}, 7$				$8, \boxed{6}$
c_5		$1, 3, 4$		$2, 5, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$\boxed{9}, 7, 8$

Choose t_2, $n_1 = 1$ and $n_2 = 5$:

	t_1	t_2	t_3	t_4	t_5
c_1	1	$3, 4$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$3, 4$	5	
c_4	$7, 9$				$6, 8$
c_5		$\boxed{5}, 3, 4$		$2, \boxed{1}, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

Choose t_2, $n_1 = 3$ and $n_2 = 8$:

	t_1	t_2	t_3	t_4	t_5
c_1	1	$\boxed{8}, 4$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$3, 4$	5	
c_4	$7, 9$				$6, \boxed{3}$
c_5		$3, 4, 5$		$1, 2, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

Choose t_2, $n_1 = 4$ and $n_2 = 6$:

	t_1	t_2	t_3	t_4	t_5
c_1	1	$\boxed{6}, 8$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$3, 4$	5	
c_4	$7, 9$				$3, 6$
c_5		$3, 4, 5$		$1, 2, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

Choose t_3, $n_1 = 3$ and $n_2 = 5$. Then

	t_1	t_2	t_3	t_4	t_5
c_1	1	$6, 8$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$\boxed{5}, 4$	$\boxed{3}$	
c_4	$7, 9$				$3, 6$
c_5		$3, 4, 5$		$1, 2, 6, 7$	$8, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

Choose t_5, $n_1 = 8$ and $n_2 = 2$. Then

	t_1	t_2	t_3	t_4	t_5
c_1	1	$6, 8$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$4, 5$	3	
c_4	$7, 9$				$3, 6$
c_5		$3, 4, 5$		$1, \boxed{8}, 6, 7$	$\boxed{2}, 9$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

Choose t_5, $n_1 = 9$ and $n_2 = 1$. Then

	t_1	t_2	t_3	t_4	t_5
c_1	1	$6, 8$			
c_2	$4, 5, 6$		$7, 8$		
c_3		$1, 2, 9$	$4, 5$	3	
c_4	$7, 9$				$3, 6$
c_5		$3, 4, 5$		$\boxed{9}, 6, 7, 8$	$2, \boxed{1}$
c_6			$1, 2, 3$		
c_7				$4, 5$	$7, 8, 9$

We have the following timetable with 9 time slots:

	1	2	3	4	5	6	7	8	9
c_1	t_1					t_3		t_2	
c_2				t_1	t_1	t_1	t_3	t_3	
c_3	t_2	t_2		t_3	t_3				t_2
c_4			t_5			t_5	t_1		t_1
c_5	t_5	t_5	t_2	t_2	t_2	t_4	t_4	t_4	t_4
c_6	t_3	t_3	t_3						
c_7				t_4	t_4		t_5	t_5	t_5

6.5 Chromatic Polynomial

Let G be a simple graph, and let $P_G(x)$ (or $P(x)$) be the number of x-colorings of G. The function $P_G(x)$, can be proved to be a polynomial of x, is called the *chromatic polynomial* of G.

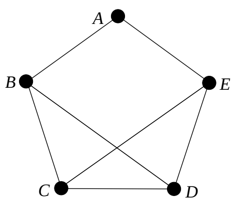
Example 6.5.1: It is easy to see that $P_{K_p}(x) = x(x-1)\cdots(x-p+1)$. $P_{N_p}(x) = x^p$. $P_{P_p}(x) = x(x-1)^{p-1}$. In general, if T is any tree with p vertices, then $P_T(k) = x(x-1)^{p-1}$.

For convenience, we let $(x)_p = P_{K_p}(x) = x(x-1)\cdots(x-p+1)$, which is called the *fall factorial* in x. For a graph G, let $f_G(r)$ (or simply $f(r)$) be the number of ways of partitioning $V(G)$ into r nonempty independent subsets.

Theorem 6.5.2: *Keep the notation above, we have* $P_G(x) = \sum\limits_{r=1}^{p} f_G(r)(x)_r$*, where* p *is the order of* G.

Corollary 6.5.3: $P_G(x)$ *is a monic polynomial of* x.

Example 6.5.4: Consider the following graph.



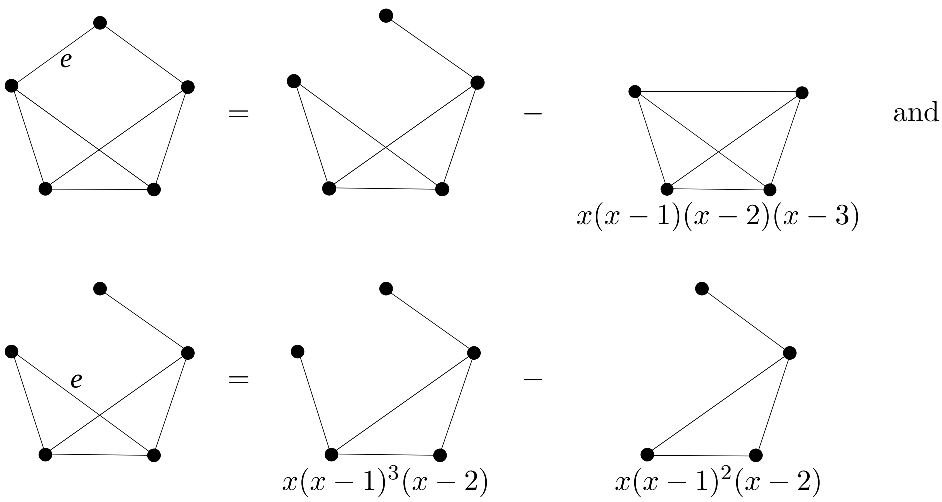
The vertex set $V = \{A, B, C, D, E\}$. Clearly $f(1) = 0$, $f(2) = 0$ and $f(5) = 1$. There are only two ways to partition V into 3 nonempty independent subsets, namely $\{\{A, D\}, \{B, E\}, \{C\}\}$ and $\{\{A, C\}, \{B, E\}, \{D\}\}$, so $f(3) = 2$. Moreover, there are only three ways to partition V into 4 nonempty independent subsets, namely $\{\{A\}, \{B, E\}, \{C\}, \{D\}\}$, $\{\{A, D\}, \{B\}, \{C\}, \{E\}\}$ and

$\{\{A, C\}, \{B\}, \{D\}, \{E\}\}$. Thus $f(4) = 3$. Therefore,

$$\begin{aligned} P_G(x) &= 2(x)_3 + 3(x)_4 + (x)_5 \\ &= 2x(x-1)(x-2) + 3x(x-1)(x-2)(x-3) + x(x-1)(x-2)(x-3)(x-4) \\ &= x(x-1)(x-2)[2 + 3(x-3) + (x-3)(x-4)] \\ &= x(x-1)(x-2)(x^2 - 4x + 5). \end{aligned}$$

Theorem 6.5.5: *Let G be a simple graph and let $e \in E(G)$. Then $P_G(x) = P_{G-e}(x) - P_{G/e}(x)$.*

Example 6.5.6: Consider the graph in Example 6.5.4 again. We have



It follows that
$P_G(x) = x(x-1)(x-2)(x^2 - 4x + 5)$. Since $P_G(1) = 0$, $P_G(2) = 0$ and $P_G(3) \neq 0$, we have $\chi(G) = 3$.