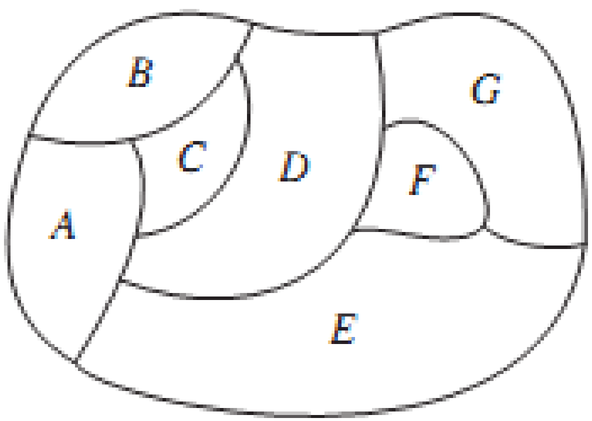
Graph coloring

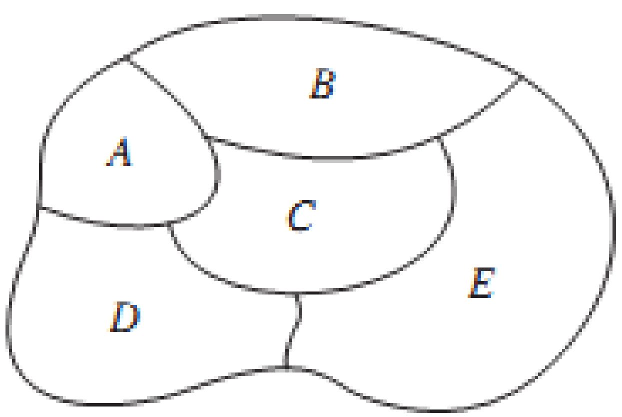
Introduction

Problems related to the coloring of maps of regions, such as maps of parts of the world, have generated many results in graph theory.

When a map is colored, 2 regions with a common border are customarily assigned different colors.

One way to ensure that 2 adjacent regions never have the same color is to use a different color for each region.





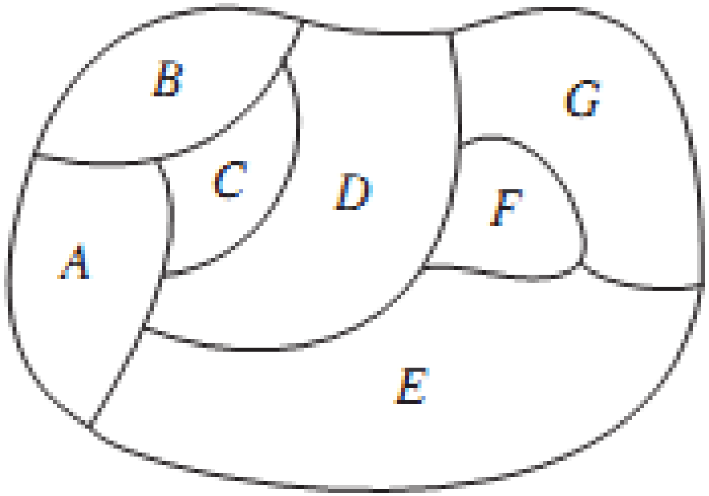
However, this is inefficient, and on maps with many regions it would be hard to distinguish similar colors.

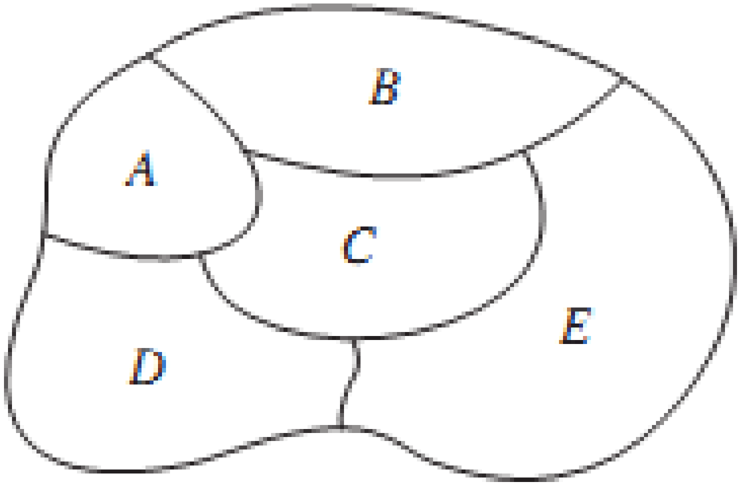
Instead, a small number of colors should be used whenever possible.

Consider the problem of determining the least number of colors that can be used to color a map so that adjacent regions never have the same color.

For instance, for the map shown on the left bellow, 4 colors suffice, but 3 colors are not enough.

In the map on the right bellow, 3 colors are sufficient (but 2 are not).





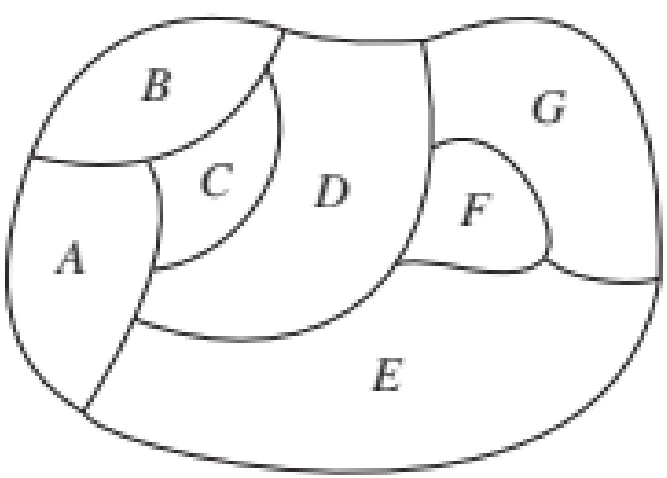
Each map in the plane can be represented by a graph.

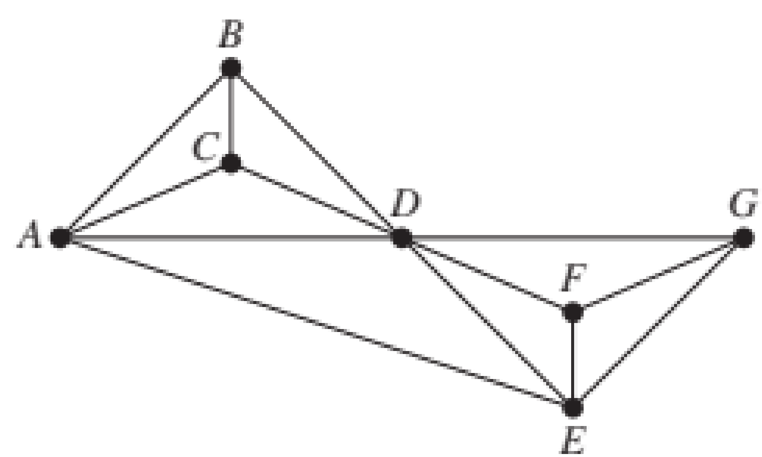
To set up this correspondence, each region of the map is represented by a vertex.

Edges connect 2 vertices if the regions represented by these vertices have a common border.

Two regions that touch at only one point are not considered adjacent.

The resulting graph is called the dual graph (двойственный граф) of the map.

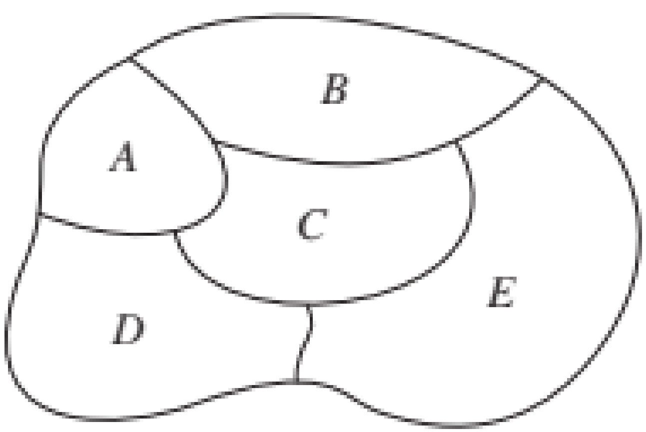


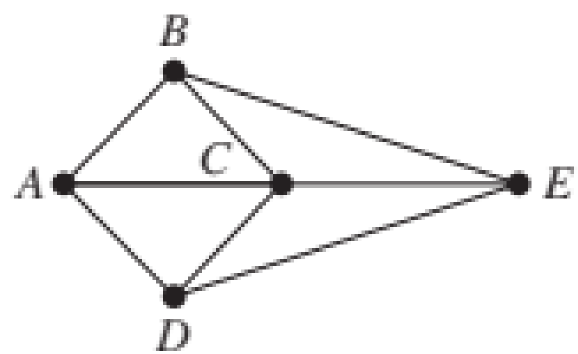


A map and the dual graph that corresponds to this map

By the way in which dual graphs of maps are constructed, it is clear that any map in the plane has a planar dual graph.

The problem of coloring the regions of a map is equivalent to the problem of coloring the vertices of the dual graph so that no two adjacent vertices in this graph have the same color.





DEFINITION 1

A proper vertex coloring, coloring (правильная вершинная раскраска, раскраска) of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color.

A graph can be colored by assigning a different color to each of its vertices.

However, for most graphs a coloring can be found that uses fewer colors than the number of vertices in the graph.

What is the least number of colors necessary?

DEFINITION 2 The chromatic number (хроматическое число) of a graph is the least number of colors needed for a a proper vertex coloring of this graph.

The chromatic number of a graph *G* is denoted by $\chi(G)$.

Note that asking for the chromatic number of a planar graph is the same as asking for the minimum number of colors required to color a planar map so that no two adjacent regions are assigned the same color.

This question has been studied > 100 years.

The answer is provided by one of the most famous theorems in mathematics.

THEOREM 1 THE FOUR COLOR THEOREM The chromatic number of a planar graph ≤ 4.

The postmark on University of Illinois mail after the Four Color Theorem was proved.

The four color theorem was originally posed as a conjecture in the 1850s.

It was finally proved by the American mathematicians Kenneth Appel and Wolfgang Haken in 1976.

Prior to 1976, many incorrect proofs were published, often with hard-to-find errors.

In addition, many futile attempts were made to construct counterexamples by drawing maps that require > 4 colors.

Perhaps the most notorious fallacious proof in all of mathematics is the incorrect proof of the four color theorem published in 1879 by a London barrister and amateur mathematician, Alfred Kempe.

Mathematicians accepted his proof as correct until 1890, when Percy Heawood found an error that made Kempe’s argument incomplete.

However, Kempe's line of reasoning turned out to be the basis of the successful proof given by Appel and Haken. Their proof relies on a careful case-by-case analysis carried out by computer.

They showed that if the four color theorem were false, there would have to be a counterexample of one of approximately 2000 different types, and they then showed that none of these types exists.

They used over 1000 hours of computer time in their proof.

This proof generated a large amount of controversy, because computers played such an important role in it.

For example, could there be an error in a computer program that led to incorrect results?

Was their argument really a proof if it depended on what could be unreliable computer output?

Since their proof appeared, simpler proofs that rely on checking fewer types of possible counterexamples have been found and a proof using an automated proof system has been created.

However, no proof not relying on a computer has yet been found.

Note that the four color theorem applies only to planar graphs.

Nonplanar graphs can have arbitrarily large chromatic numbers, as will be shown in Example 2.

Two things are required to show that the chromatic number of a graph is k.

1) We must show that the graph can be colored with k colors.

This can be done by constructing such a coloring.

2) We must show that the graph cannot be colored using $< k$ colors.

EXAMPLE 1

What are the chromatic numbers of the graph *G*?

Solution: The chromatic number of *G* is ≥ 3, because the vertices *a*, *b*, and *c* must be assigned different colors.

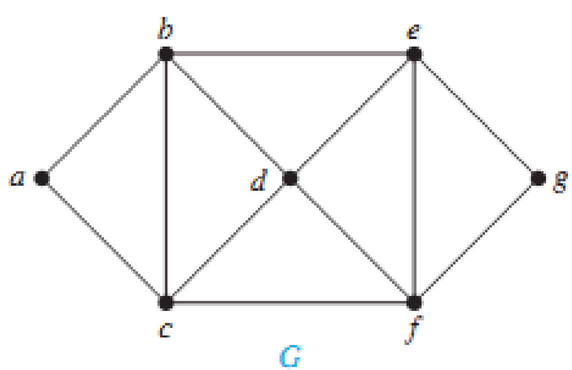
To see if *G* can be colored with 3 colors, assign red to *a*, blue to *b*, and green to *c*.

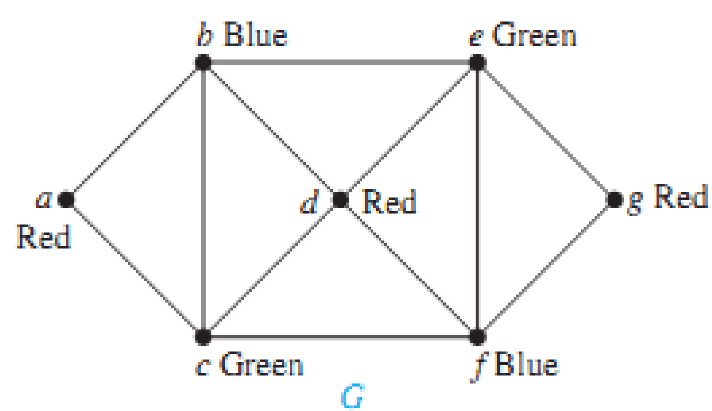
Then, *d* can (and must) be colored red because it is adjacent to *b* and *c*.

Furthermore, *e* can (and must) be colored green because it is adjacent only to vertices colored red and blue, and *f* can (and must) be colored blue because it is adjacent only to vertices colored red and green.

Finally, *g* can (and must) be colored red because it is adjacent only to vertices colored blue and green.

This produces a coloring of *G* using exactly 3 colors.



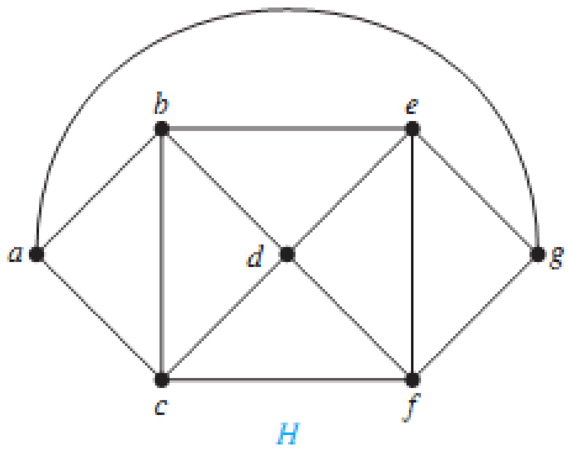


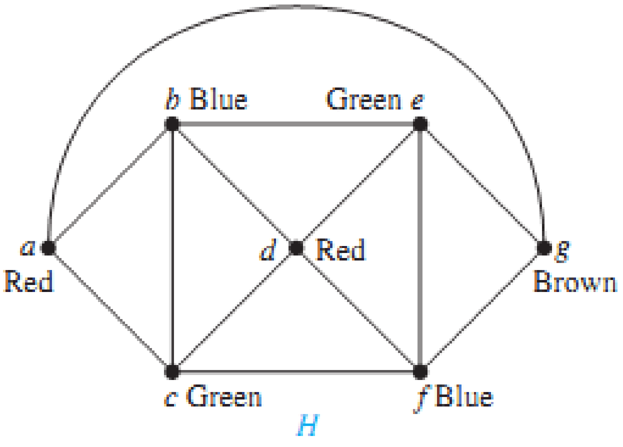
The graph *H* is made up of the graph *G* with an edge connecting *a* and *g*.

Any attempt to color *H* using 3 colors must follow the same reasoning as that used to color *G*, except at the last stage, when all vertices other than *g* have been colored.

Then, because *g* is adjacent (in *H*) to vertices colored red, blue, and green, a fourth color, say brown, needs to be used.

Hence, *H* has a chromatic number equal to 4. A coloring of *H* is shown bellow.





EXAMPLE 2

What is the chromatic number of K_n?

Solution: A coloring of K_n can be constructed using n colors by assigning a different color to each vertex.

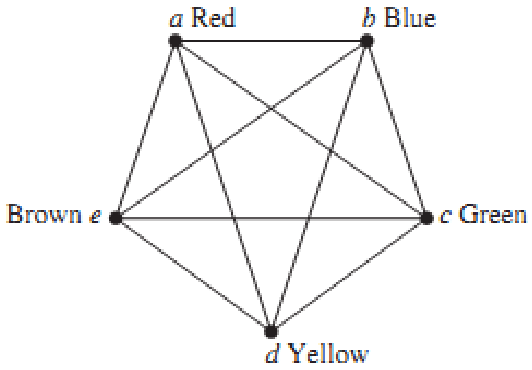
Is there a coloring using fewer colors? The answer is no.

No two vertices can be assigned the same color, because every two vertices of this graph are adjacent.

Hence, the chromatic number of K_n is n. That is, $\chi(K_n) = n$.

(Recall that K_n is not planar when $n \geq 5$, so this result does not contradict the four color theorem.)

A coloring of K_5 using 5 colors is shown bellow.



EXAMPLE 3

What is the chromatic number of the complete bipartite graph $K_{m,n}$, where *m* and *n* are positive integers?

Solution: The number of colors needed may seem to depend on *m* and *n*.

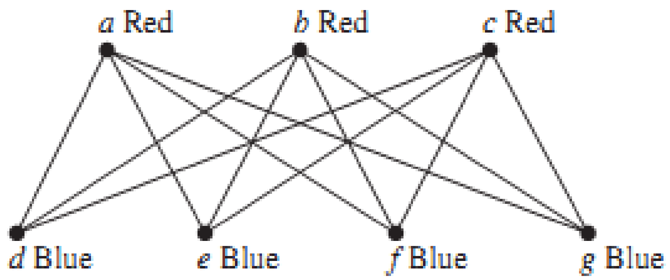
However, we know that only 2 colors are needed, because $K_{m,n}$ is a bipartite graph.

Hence, $\chi(K_{m,n}) = 2$.

This means that we can color the set of *m* vertices with one color and the set of *n* vertices with a second color.

Because edges connect only a vertex from the set of m vertices and a vertex from the set of n vertices, no two adjacent vertices have the same color.

A coloring of $K_{3,4}$ with two colors is displayed bellow.



EXAMPLE 4 What is the chromatic number of the graph C_n, where $n \geq 3$?

Solution: We will first consider some individual cases.

To begin, let $n = 6$. Pick a vertex and color it red.

Proceed clockwise in the planar depiction of C_6 shown bellow.

It is necessary to assign a second color, say blue, to the next vertex reached.

Continue in the clockwise direction;

the 3rd vertex can be colored red,

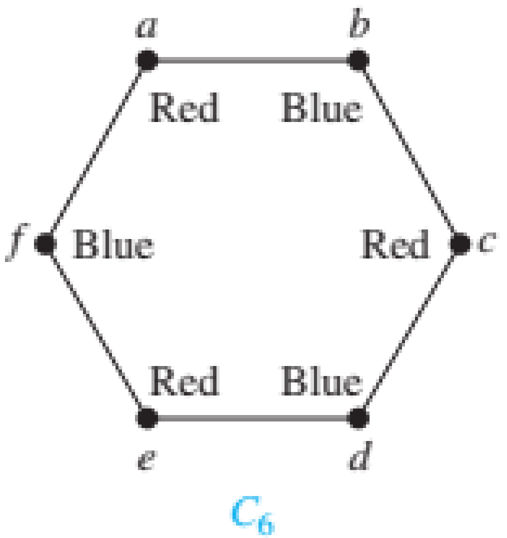
the 4th vertex blue,

and the 5th vertex red.

Finally, the 6th vertex, which is adjacent to the first, can be colored blue.

Hence, the chromatic number of C_6 is 2.

The coloring constructed here is shown bellow.



Coloring of C_5

Next, let $n = 5$ and consider C_5.

Pick a vertex and color it red.

Proceeding clockwise, it is necessary to assign a second color, say blue, to the next vertex reached.

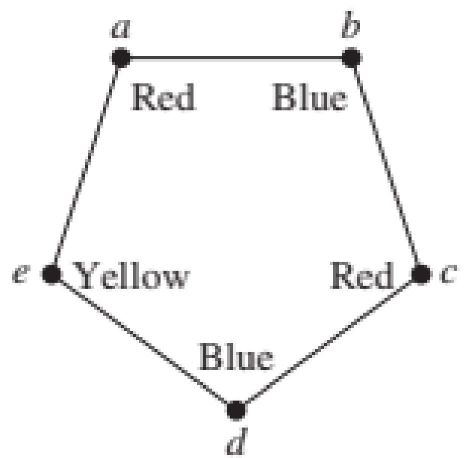
Continuing in the clockwise direction, the third vertex can be colored red, and the fourth vertex can be colored blue.

The fifth vertex cannot be colored either red or blue, because it is adjacent to the fourth vertex and the first vertex.

Consequently, a third color is required for this vertex.

Note that we would have also needed 3 colors if we had colored vertices in the counterclockwise direction.

Thus, the chromatic number of C_5 is 3.



C_5

In general, 2 colors are needed to color C_n when n is even.

To construct such a coloring, simply pick a vertex and color it red.

Proceed around the graph in a clockwise direction (using a planar representation of the graph) coloring the second vertex blue, the third vertex red, and so on.

The nth vertex can be colored blue, because the two vertices adjacent to it, namely the(n − 1)st and the first vertices, are both colored red.

When n is odd and n> 1, the chromatic number of C_n is 3.

To see this, pick an initial vertex.

To use only 2 colors, it is necessary to alternate colors as the graph is traversed in a clockwise direction.

However, the nth vertex reached is adjacent to two vertices of different colors, namely, the first and (n − 1)st.

Hence, a third color must be used.

We have shown that $\chi(C_n) = 2$ if n is an even positive integer with $n \geq 4$ and $\chi(C_n) = 3$ if n is an odd positive integer with $n \geq 3$.

Applications

Colouring problems arise naturally in many practical situations where it is required to partition a set of objects into groups in such a way that the members of each group are mutually compatible according to some criterion.

Scheduling Final Exams

How can the final exams at a university be scheduled so that no student has 2 exams at the same time?

Solution: This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between 2 vertices if there is a common student in the courses they represent.

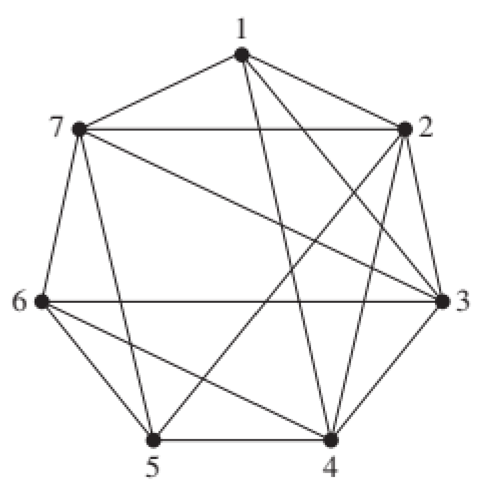
Each time slot for a final exam is represented by a different color.

A scheduling of the exams corresponds to a coloring of the associated graph.

For instance, suppose there are 7 finals to be scheduled.

The courses are numbered 1 through 7.

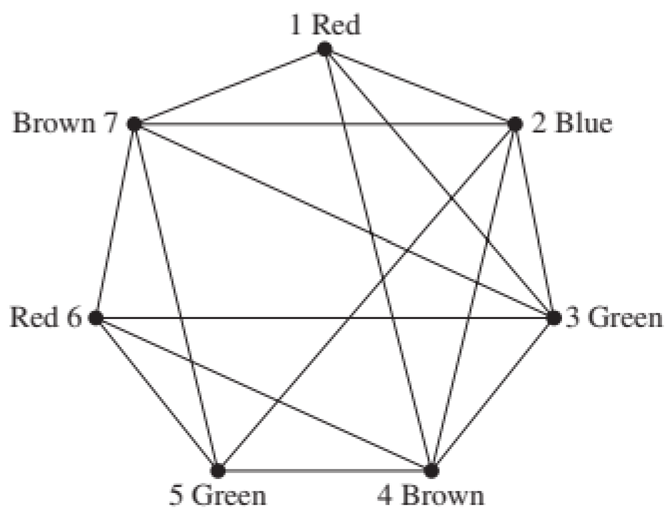
Suppose that the following pairs of courses have common students: 1 and 2, 1 and 3, 1 and 4, 1 and 7, 2 and 3, 2 and 4, 2 and 5, 2 and 7, 3 and 4, 3 and 6, 3 and 7, 4 and 5, 4 and 6, 5 and 6, 5 and 7, and 6 and 7.



The graph associated with this set of classes

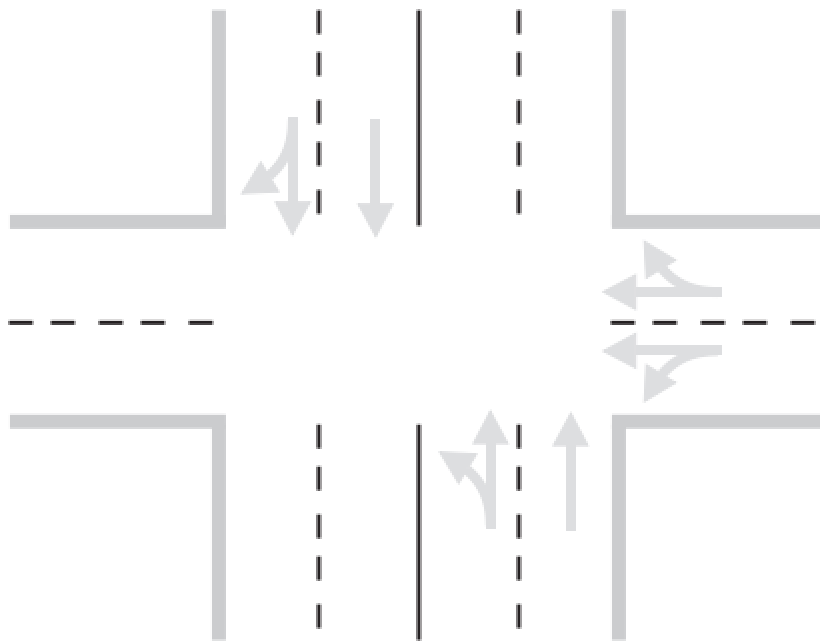
A scheduling consists of a coloring of this graph.
Because the chromatic number of this graph is 4, so 4 time slots are needed.

A coloring of the graph using 4 colors and the associated schedule are shown bellow.



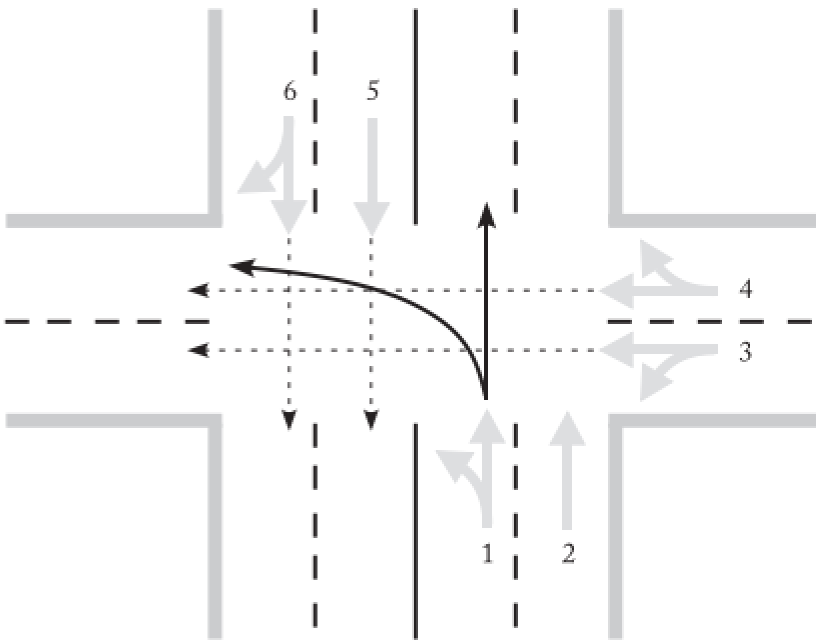
Time Period	Courses
I	1, 6
II	2
III	3, 5
IV	4, 7

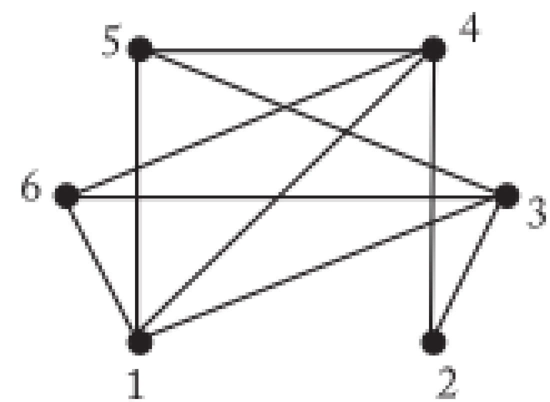
Traffic-light cycle



Problem: Find a way to allow as many lanes to have green lights as possible at the same time, while also keeping drivers from colliding with each other.

Traffic-light cycle

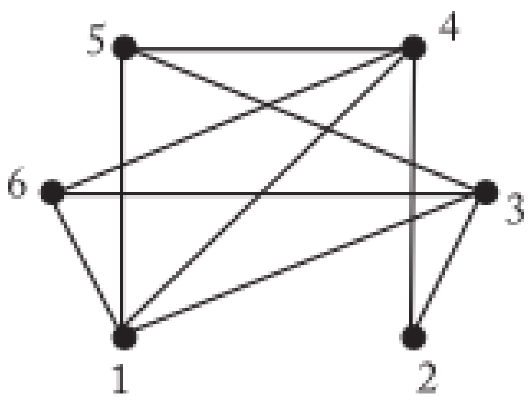


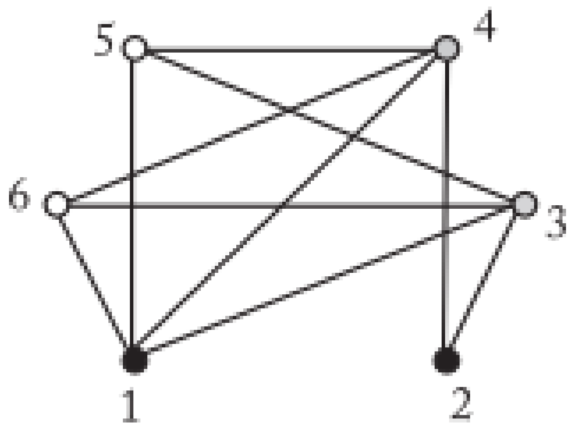


Solution: we can number the lanes

and use these numbers as vertex labels and see which paths of travel intersect to find edges in the corresponding graph.

Traffic-light cycle





Frequency Assignments

Television channels 2 through 13 are assigned to stations in North America so that no two stations within 150 miles can operate on the same channel (frequency).

How can the assignment of channels be modeled by graph coloring?

Solution: Construct a graph by assigning a vertex to each station. Two vertices are connected by an edge if they are located within 150 miles of each other.

An assignment of channels corresponds to a coloring of the graph, where each color represents a different channel.

Index Registers

In efficient compilers the execution of loops is speeded up when frequently used variables are stored temporarily in index registers in the central processing unit (CPU), instead of in regular memory.

For a given loop, how many index registers are needed?

This problem can be addressed using a graph coloring model.

To set up the model, let each vertex of a graph represent a variable in the loop.

There is an edge between two vertices if the variables they represent must be stored in index registers at the same time during the execution of the loop.

Thus, the chromatic number of the graph gives the number of index registers needed, because different registers must be assigned to variables when the vertices representing these variables are adjacent in the graph.

A natural question: What is the relation between the chromatic number of a graph G and chromatic number of a subgraph of G?

THEOREM 2 If H is a subgraph of G, $\chi(H) \le \chi(G)$.

Proof. Any coloring of G provides a proper coloring of H, simply by assigning the same colors to vertices of H that they have in G.

This means that H can be colored with $\chi(G)$ colors, perhaps even fewer, which is exactly what we want.

Often this fact is interesting "in reverse".

For example, if G has a subgraph H that is a complete graph K_m, then $\chi(H) = m$ and so $\chi(G) \ge m$.

A subgraph of G that is a complete graph is called a clique, and there is an associated graphical parameter.

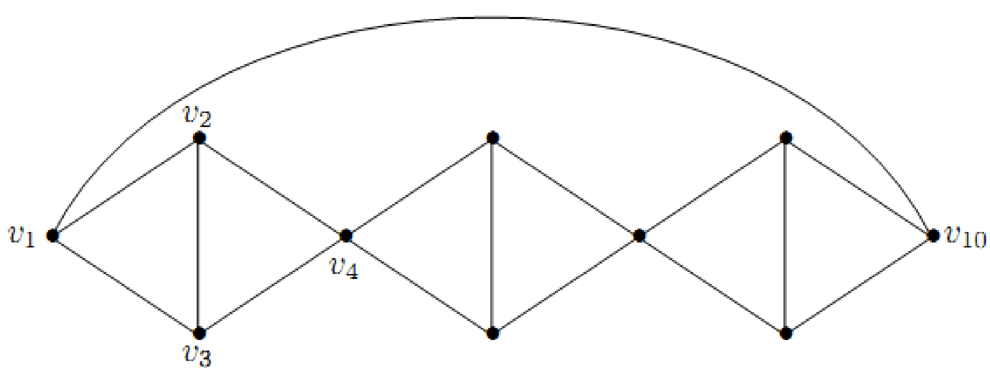
Clique number (кликовое число)

DEFINITION 2 The clique number of a graph *G* is the largest *m* such that K_m is a subgraph of *G*.

It is tempting to speculate that the only way a graph *G* could require *m* colors is by having such a subgraph.

This is false; graphs can have high chromatic number while having low clique number; see figure bellow.

It is easy to see that this graph has $\chi \geq 3$, because there are many 3-cliques in the graph.



Clique number (кликовое число)

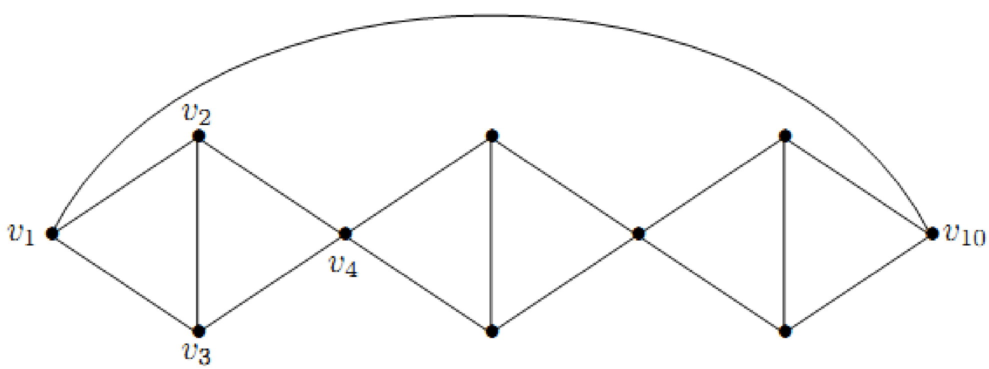
In general it can be difficult to show that a graph cannot be colored with a given number of colors, but in this case it is easy to see that the graph cannot in fact be colored with 3 colors, because so much is “forced”.

Suppose the graph can be colored with 3 colors.

Starting at the left if vertex v_1 gets color 1, then v_2 and v_3 must be colored 2 and 3, and vertex v_4 must be color 1.

Continuing, v_{10} must be color 1, but this is not allowed, so $\chi > 3$.

On the other hand, since v_{10} can be colored 4, we see $\chi = 4$.



Paul Erd˝os showed in 1959 that there are graphs with arbitrarily large chromatic number and arbitrarily large girth (обхват) (the girth is the size of the smallest cycle in a graph).

This is much stronger than the existence of graphs with high chromatic number and low clique number.

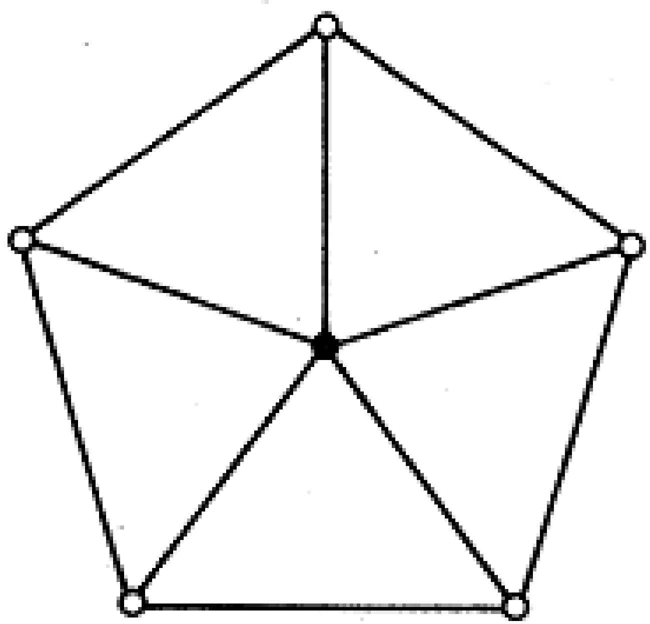
Independent set (независимое множество)

Graph coloring is closely related to the concept of an independent set.

DEFINITION 3 A set *S* of vertices in a graph is independent if no two vertices of *S* are adjacent.

It is easy to find independent sets: just pick vertices that are mutually non-adjacent.

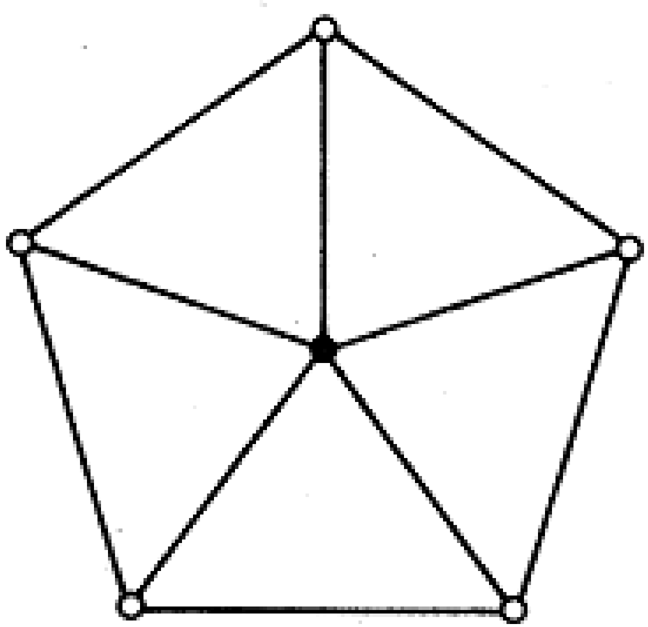
A single vertex set, for example, is independent.

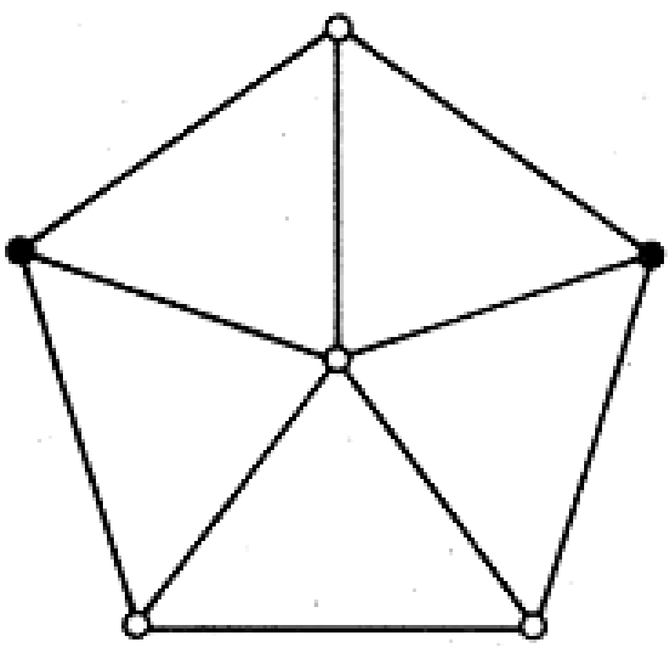


Independence number (число независимости)

If a graph is properly colored, the vertices that are assigned a particular color form an independent set.

The interesting quantity is the maximum size of an independent set. The independence number of G is the maximum size of an independent set; it is denoted $\alpha(G)$.





Chromatic number and independence number

The natural first question about these graphical parameters is: how small or large can they be in a graph G with n vertices.

It is easy to see that

$1 \leq \chi(G) \leq n$

$1 \leq \alpha(G) \leq n$

and that the limits are all attainable:

A graph with no edges has chromatic number 1 and independence number n:

$\chi(O_n) = 1$, $\alpha(O_n) = n$.

A complete graph has chromatic number n and independence number 1:

$\chi(K_n) = n$, $\alpha(K_n) = 1$.

These inequalities are thus not very interesting.

Color class (цветной класс)

A color class (цветной класс) is the set of all vertices of a single color.

If a graph is properly colored, then each color class is an independent set.

THEOREM 3 In any graph G on n vertices,
$n/\alpha \leq \chi$.

Proof. Suppose G is colored with χ colors.
Since each color class is independent, the size of any color class is $\leq \alpha$.
Let the color classes be $V_1, V_2, \ldots, V_\chi$.
Then

$$n = \sum_{i=1}^{\chi} |V_i| \leq \chi\alpha,$$

as desired.

We can improve the upper bound on $\chi(G)$ as well.

Let $\Delta(G)$ denotes the maximum degree of any vertex.

THEOREM 4 In any graph G, $\chi \leq \Delta + 1$.

Proof. We show that we can always color G with $\Delta + 1$ colors by a simple greedy algorithm:

Pick a vertex v_n, and list the vertices of G as $v_1, v_2, \ldots, v_n$ so that if $i < j$, $d(v_i, v_n) \geq d(v_j, v_n)$,

that is, we list the vertices farthest from v_n first.

We use integers $1, 2, \ldots, \Delta + 1$ as colors.

Color v_1 with 1.

Then for each v_i in order, color v_i with the smallest integer that does not violate the proper coloring requirement, that is, which is different than the colors already assigned to the neighbors of v_i.

For $i < n$, we claim that v_{i} is colored with one of 1, 2,..., Δ.

This is certainly true for v_1.

For $1 < i < n$, v_i has at least one neighbor that is not yet colored, namely, a vertex closer to v_{n} on a shortest path from v_n to v_i.

Thus, the neighbors of v_i use $\leq \Delta - 1$ colors from the colors 1, 2,..., Δ, leaving ≥ 1 color from this list available for v_i.

Once v_1, ..., $v_{\text{n}-1}$ have been colored, all neighbors of v_n have been colored using the colors 1, 2, . . . , Δ, so color $\Delta + 1$ may be used to color v_n.

Note that if $\deg(v_n) < \Delta$, even v_n may be colored with one of the colors $1, 2, \ldots, \Delta$.

Since the choice of v_n was arbitrary, we may choose v_n so that $\deg(v_n) < \Delta$, unless all vertices have degree Δ, that is, if G is regular.

Thus, we have proved somewhat more than stated, namely, that any graph G that is not regular has $\chi \leq \Delta$.

(If instead of choosing the particular order of $v_1, \ldots, v_n$ that we used we were to list them in arbitrary order, even vertices other than v_n might require use of color $\Delta+1$.

This gives a slightly simpler proof of the stated theorem.) We state this as a corollary.

COROLLARY 5. If G is not regular, $\chi \leq \Delta$.

There are graphs for which $\chi = \Delta + 1$:

any cycle of odd length has $\Delta = 2$ and $\chi = 3$,

and K_n has $\Delta = n - 1$ and $\chi = n$.

Of course, these are regular graphs.

It turns out that these are the only examples, that is, if G is not an odd cycle or a complete graph, then $\chi(G) \leq \Delta(G)$.

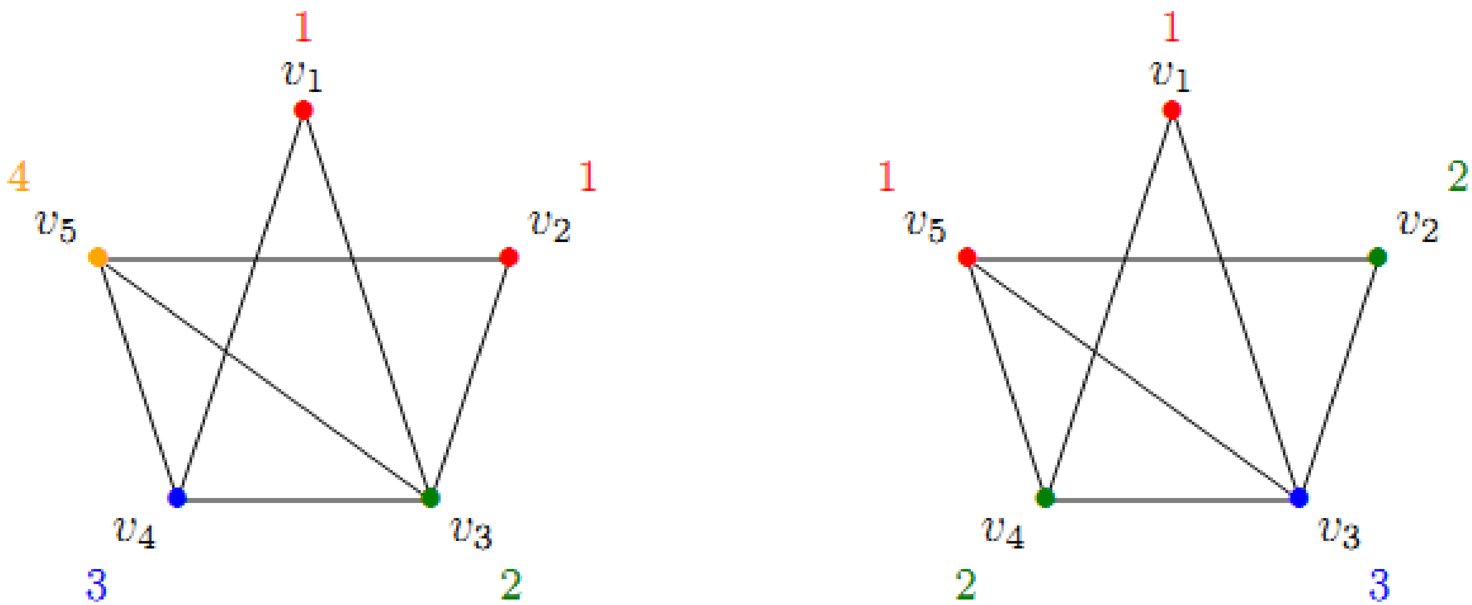
THEOREM 6 (Brooks's 1941)

If G is a graph other than K_n or C_{2n+1},

$\chi \leq \Delta$.

The greedy algorithm will not always color a graph with the smallest possible number of colors.

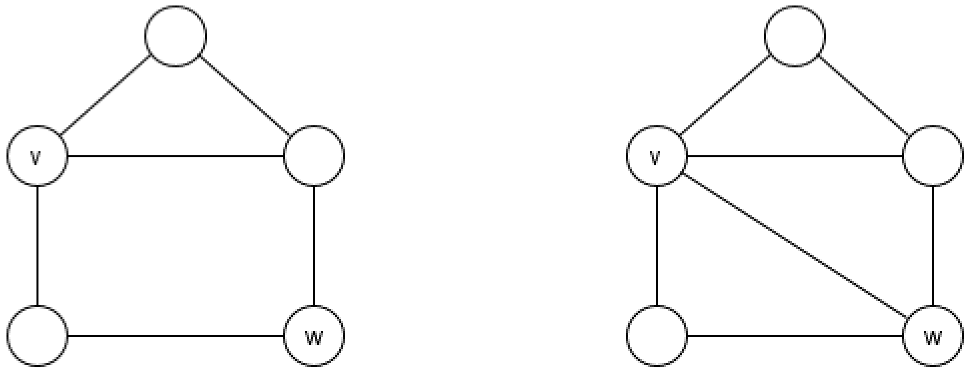
Figure bellow shows a graph with chromatic number 3, but the greedy algorithm uses 4 colors if the vertices are ordered as shown.



In general, it is difficult to compute $\chi(G)$, that is, it takes a large amount of computation, but there is a simple algorithm for graph coloring that is not fast.

Suppose that v and w are non-adjacent vertices in G.

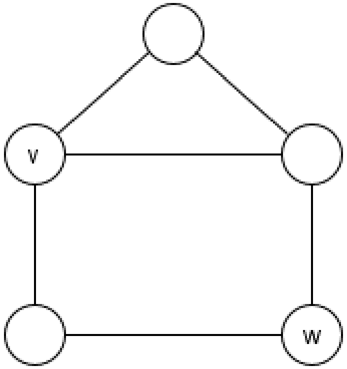
Denote by $G + \{v, w\} = G + e$ the graph formed by adding edge $e = \{v, w\}$ to G.



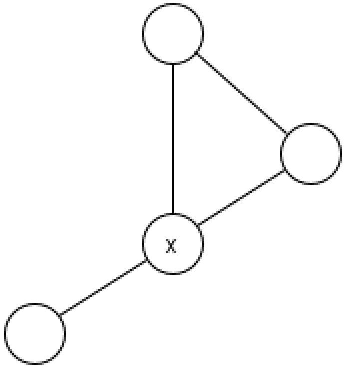
G $\qquad\qquad$ $G + e$

Denote by *G/e* the graph in which *v* and *w* are "identified", that is, *v* and *w* are replaced by a single vertex *x* adjacent to all neighbors of *v* and *w*.

(But note that we do not introduce multiple edges: if *u* is adjacent to both *v* and *w* in *G*, there will be a single edge from *x* to *u* in *G/e*.)



G

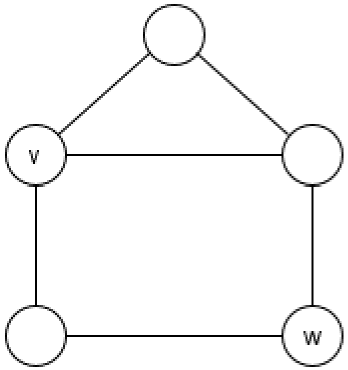


G/e

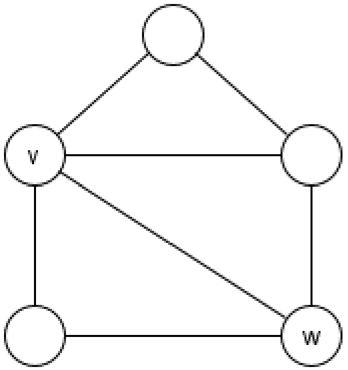
Consider a proper coloring of *G* in which *v* and *w* are different colors; then this is a proper coloring of *G+e* as well.

Also, any proper coloring of *G+e* is a proper coloring of *G* in which *v* and *w* have different colors.

So a coloring of *G+e* with the smallest possible number of colors is a best coloring of *G* in which *v* and *w* have different colors, that is, $\chi(G + e)$ is the smallest number of colors needed to color *G* so that *v* and *w* have different colors.



G

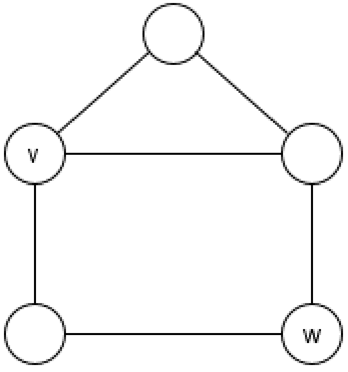


G+e

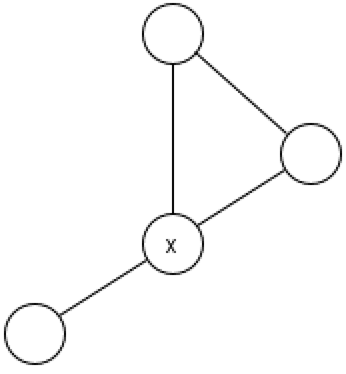
If *G* is properly colored and *v* and *w* have the same color, then this gives a proper coloring of *G*/*e*, by coloring *x* in *G*/*e* with the same color used for *v* and *w* in *G*.

Also, if *G/e* is properly colored, this gives a proper coloring of *G* in which *v* and *w* have the same color, namely, the color of *x* in *G*/e.

Thus, $\chi(G/e)$ is the smallest number of colors needed to properly color *G* so that *v* and *w* are the same color.



G



G/e

The upshot of these observations is that
$\chi(G) = \min(\chi(G+e), \chi(G/e))$.

This algorithm can be applied recursively,
that is, if $G_1 = G+e$ and $G_2 = G/e$

then $\chi(G_1) = \min(\chi(G_1 + e), \chi(G_1/e))$ and

$\chi(G_2) = \min(\chi(G_2 + e), \chi(G_2/e))$,

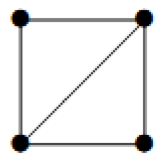
where of course the edge e is different in each graph.

Continuing in this way, we can eventually compute $\chi(G)$, provided that eventually we end up with graphs that are "simple" to color.

Roughly speaking, because G/e has fewer vertices, and $G + e$ has more edges, we must eventually end up with a complete graph along all branches of the computation.

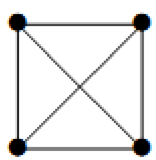
Whenever we encounter a complete graph K_m it has chromatic number m, so no further computation is required along the corresponding branch.

EXAMPLE 5. We illustrate with a very simple graph:



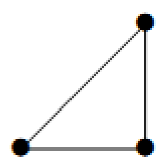
3

G



4

G + *e*



3

G/*e*

The chromatic number of the graph at the top is min(3, 4) = 3. (Of course, this is quite easy to see directly.)

Let’s make this more precise.

THEOREM 7. The algorithm above correctly computes the chromatic number in a finite amount of time.

Proof. Suppose that a graph G has n vertices and m edges. The number of pairs of non-adjacent vertices is

$na(G) = n(n\text{-}1)/2 - m$.

The proof is by induction on na.

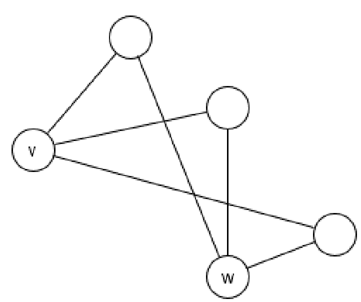
a) If $na(G) = 0$ then G is a complete graph and the algorithm terminates immediately.

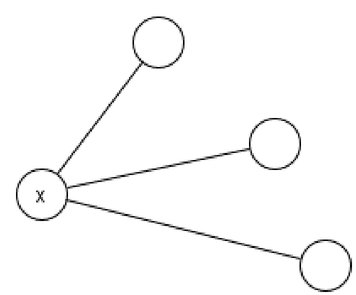
b) Now we note that $na(G + e) < na(G)$ and $na(G/e) < na(G)$:

$$\mathrm{na}(G+e) = \binom{n}{2} - (m+1) = \mathrm{na}(G) - 1$$

$$\mathrm{na}(G/e) = \binom{n-1}{2} - (m-c),$$

where *c* is the number of neighbors that *v* and *w* have in common.





$$
\begin{aligned}
\mathrm{na}(G/e) &= \binom{n-1}{2} - m + c \\
&\leq \binom{n-1}{2} - m + n - 2 \\
&= \frac{(n-1)(n-2)}{2} - m + n - 2 \\
&= \frac{n(n-1)}{2} - \frac{2(n-1)}{2} - m + n - 2 \\
&= \binom{n}{2} - m - 1 \\
&= \mathrm{na}(G) - 1.
\end{aligned}
$$

Now if $na(G) > 0$, G is *not a complete graph*, so there are non-adjacent vertices v and w.

By the induction hypothesis the algorithm computes $\chi(G + e)$ and $\chi(G/e)$ correctly, and finally it computes $\chi(G)$ from these in one additional step.

While this algorithm is very inefficient, it is sufficiently fast to be used on small graphs with the aid of a computer.

The Chromatic Polynomial

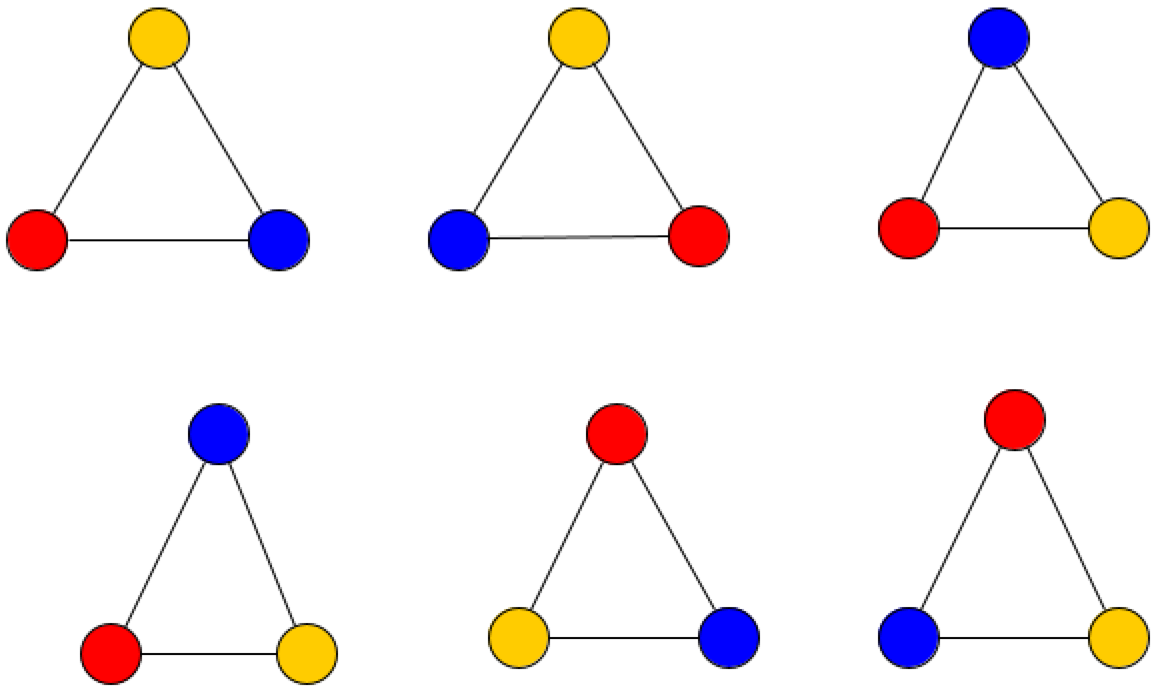
We now turn to the number of ways to color a graph G with k colors.

Two colourings are to be regarded as distinct if some vertex is assigned different colours in the two colourings;

in other words, if $\{V_1, V_2, \ldots, V_k\}$ and $\{V'_1, V'_2, \ldots, V'_k\}$ are two k-colourings,

then $\{V_1, V_2, \ldots, V_k\} = \{V'_1, V'_2, \ldots, V'_k\} \Leftrightarrow V_i = V'_i$ for $1 \leq i \leq k$.

A triangle, for example, has 6 distinct 3-colourings.



EXAMPLE 6. If G is K_n,

$P_G(k) = k(k-1)(k-2) \cdot\cdot\cdot (k-n+1)$, $k \geq n$

namely, the number of permutations of k things taken n at a time.

Vertex 1 may be colored any of the k colors, vertex 2 any of the remaining $k - 1$ colors, and so on.

When $k < n$, $P_G(k) = 0$.

EXAMPLE 7 If G has n vertices and no edges,

$P_G(k) = k^n$.

Each vertex can be independently assigned any one of the available colors.

We can provide an easy mechanical procedure for the computation of $P_G(k)$, quite similar to the algorithm we presented for computing $\chi(G)$.

Suppose G has edge $e = \{v, w\}$, and consider $P_{G-e}(k)$, the number of ways to color $G - e$ with k colors.

Theorem 8. If G is simple then

$P_{G-e}(k) = P_G(k) + P_{G/e}(k)$

Proof Let u and v be the ends of e.

To each k - coloring of $G - e$ that assigns the same color to u and v, there corresponds a k -coloring of G/e in which the vertex of G/e formed by identifying u and v is assigned the common color of u and v.

Therefore $P_{G/e}(k)$ is precisely the number of k-colorings of $G - e$ in which u and v are assigned the same color.

Also, since each k-coloring of $G-e$ that assigns different colors to u and v is a k -coloring of G, and conversely, $P_G(k)$ is the number of k -colorings of $G - e$ in which u and v are assigned different colors.

It follows that $P_{G-e}(k) = P_G(k) + P_{G/e}(k)$

Thus,

$$P_{G\text{-}e}(k) = P_G(k) + P_{G/e}(k)$$

The formula can also be rewritten as follows

$$P_G(k) = P_{G-e}(k) - P_{G/e}(k).$$

Theorem 8 provides a means of calculating the chromatic polynomial of a graph recursively.

It can be used in either of 2 ways:

(1) by repeatedly applying the recursion

$$P_{G-e}(k) = P_G(k) + P_{G/e}(k)$$

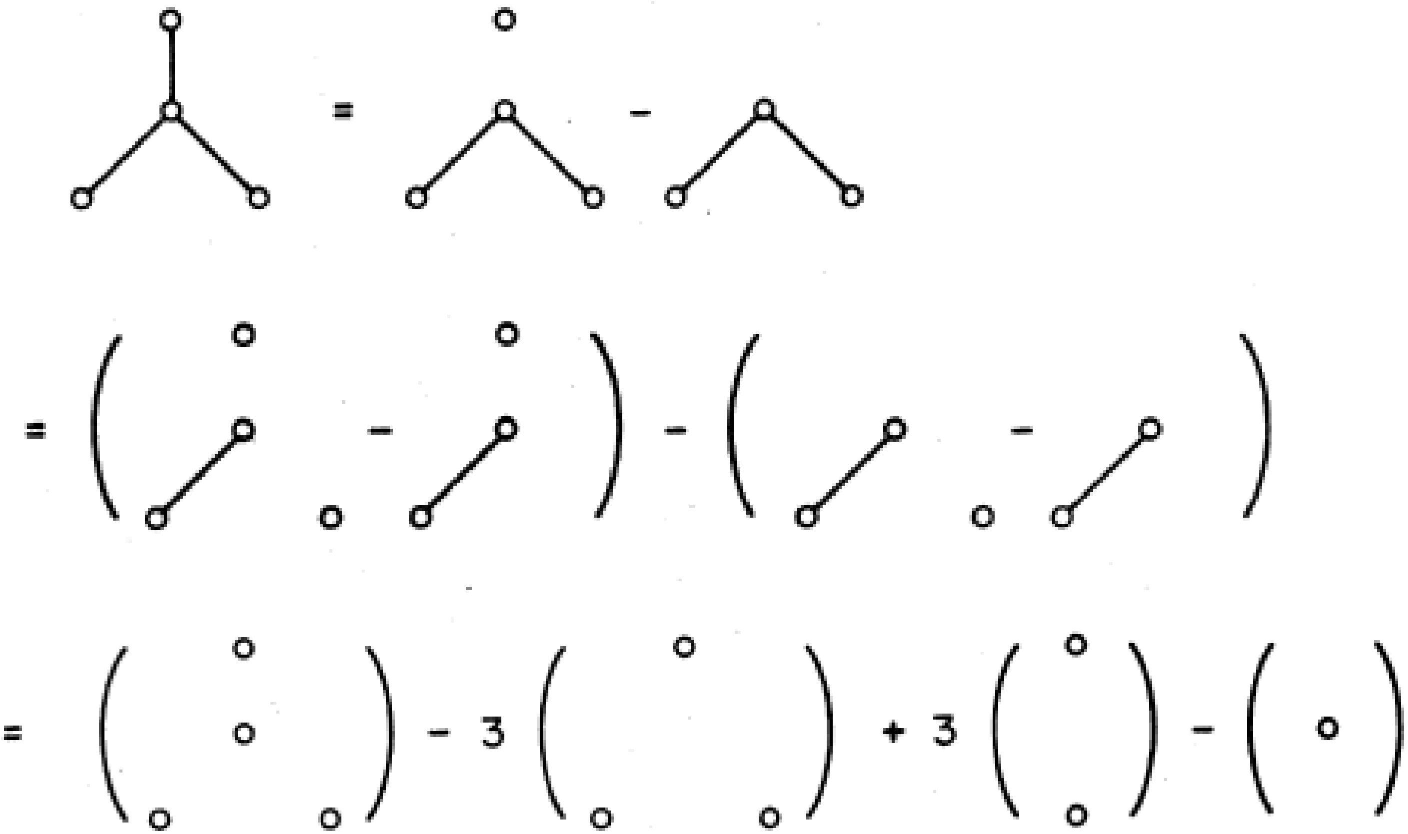
thereby expressing $P_{G-e}(k)$ as a linear combination of chromatic polynomials of complete graphs.

2) by repeatedly applying the recursion

$$P_G(k) = P_{G-e}(k) - P_{G/e}(k).$$

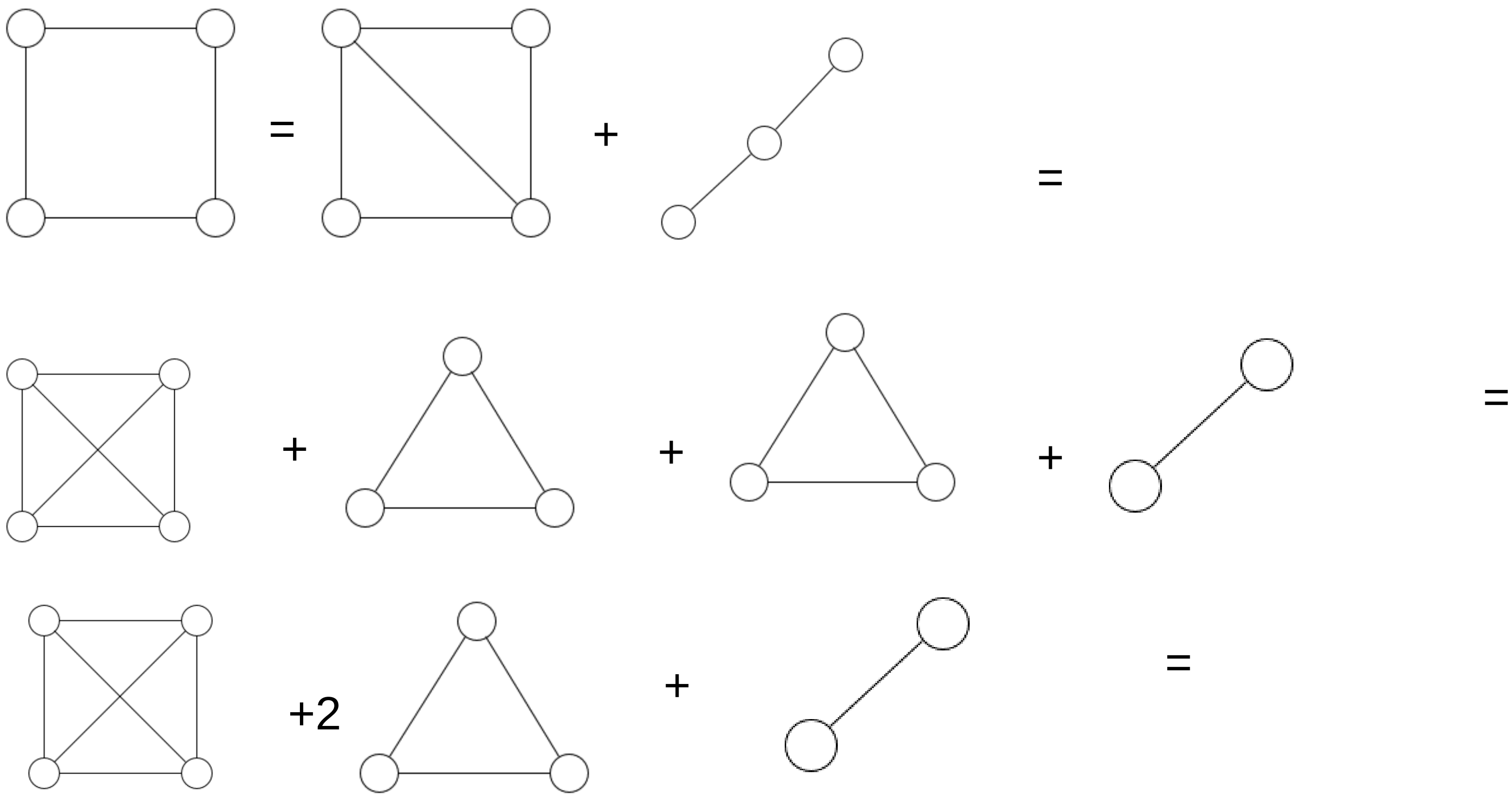
thereby expressing $P_G(k)$ as a linear combination of chromatic polynomials of empty graphs

Using empty graphs: $P_G(k) = P_{G-e}(k) - P_{G/e}(k).$



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

Using complete graphs: $P_{G-e}(k) = P_G(k) + P_{G/e}(k)$



$= k(k\text{-}1)(k\text{-}2)(k\text{-}3) + 2(k(k\text{-}1)(k\text{-}2) + k(k\text{-}1) = k(k\text{-}1)(k^2\text{-}3k+3)$

Edge colouring

A *k-edge-colouring* of a graph $G = (V, E)$ is a mapping $c : E \to S$,
where S is a set of *k colours*, in other words, an assignment of k colours to the edges of G.

Usually, the set of colours S is taken to be $\{1, 2, \ldots, k\}$.

A k-edge-colouring can then be thought of as a partition $\{E_1, E_2, \ldots, E_k\}$ of E, where E_i denotes the (possibly empty) set of edges assigned colour i.
An edge colouring is *proper* if adjacent *edges* receive distinct colours.

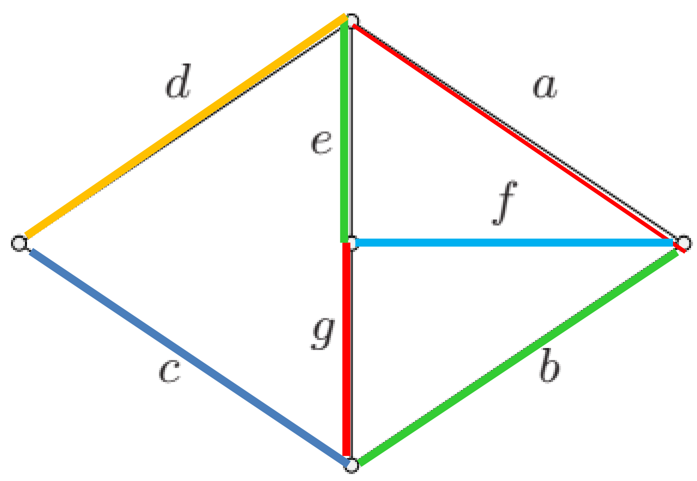
Thus a proper k-edge-colouring is a k-edge-colouring $\{M_1, M_2, \ldots, M_k\}$ in which each subset M_i is a matching.

(Because loops are self-adjacent, only loopless graphs admit proper edge colourings.)
As we are concerned here only with *proper edge colourings*, all graphs are assumed to

be loopless, and we refer to a *proper edge colouring* simply as an '*edge colouring*'.

A 4-edge-chromatic graph

The graph bellow has the 4-edge-colouring *{{a,g}, {b,e}, {c,f}, {d}}*



A graph *is k-edge-colourable* if it has *a proper k-edge-colouring*.

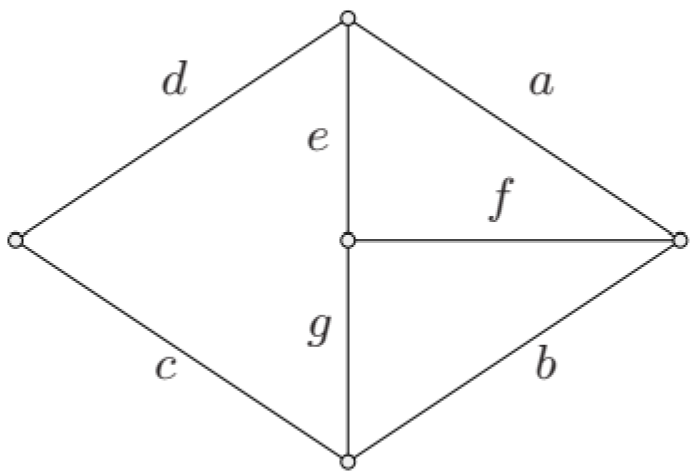
Clearly, if G is k-edge-colourable, G is also l-edge colourable for every $l > k$.

The *edge chromatic number, chromatic index (хроматический индекс)*,

$\chi'(G)$, of a graph G is the minimum k for which G is k-edge colourable,

and G is *k-edge-chromatic* if $\chi'(G) = k$.

This graph bellow is therefore 4-edge-chromatic.



In an edge colouring, the edges incident with any one vertex must evidently be assigned different colours.

This observation yields the lower bound

$\chi' \geq \Delta$

Edge colouring problems arise in practice in much the same way as do vertex colouring problems.

Theorem Vizing's Theorem

For any simple graph G, $\chi' \leq \Delta + 1$.