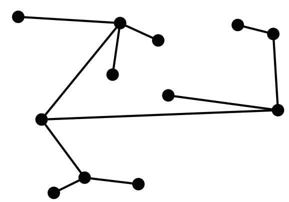
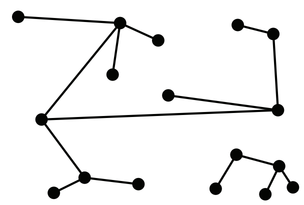
Árboles

- Un árbol es un grafo conexo y acíclico (sin ciclos).
- Un bosque es un grafo acíclico, o sea, una unión disjunta de árboles.
- Una hoja en un grafo es un vértice de grado 1.
- Un árbol generador de un grafo G es un subgrafo generador de G que es un árbol.



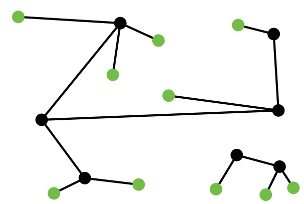
Árboles

- Un árbol es un grafo conexo y acíclico (sin ciclos).
- Un bosque es un grafo acíclico, o sea, una unión disjunta de árboles.
- Una hoja en un grafo es un vértice de grado 1.
- Un árbol generador de un grafo G es un subgrafo generador de G que es un árbol.



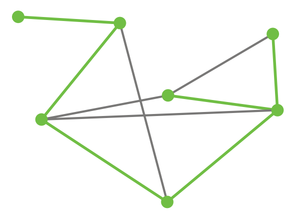
Árboles

- Un árbol es un grafo conexo y acíclico (sin ciclos).
- Un bosque es un grafo acíclico, o sea, una unión disjunta de árboles.
- Una hoja en un grafo es un vértice de grado 1.
- Un árbol generador de un grafo G es un subgrafo generador de G que es un árbol.



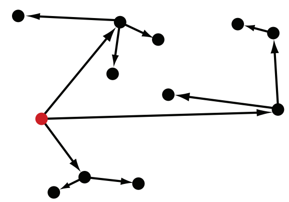
Árboles

- Un árbol es un grafo conexo y acíclico (sin ciclos).
- Un bosque es un grafo acíclico, o sea, una unión disjunta de árboles.
- Una hoja en un grafo es un vértice de grado 1.
- Un árbol generador de un grafo G es un subgrafo generador de G que es un árbol.



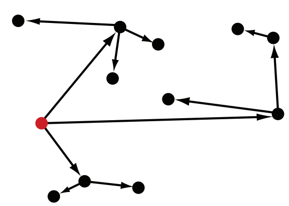
Árboles dirigidos

- Un árbol dirigido es un árbol con un vértice distinguido como raíz y tal que cada arista apunta del vértice más cercano a la raíz al más lejano.
- Si en un árbol dirigido existe la arista ij, se dice que i es el padre de j, o que j es un hijo de i.
- Obs: Todo vértice tiene un solo padre, salvo la raíz que no tiene. Se llaman hojas los vértices sin hijos.
- Se define el nivel de un vértice como su distancia a la raíz.



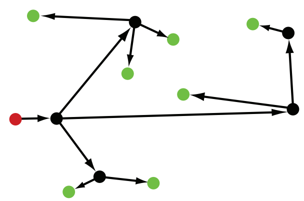
Árboles dirigidos

- Un árbol dirigido es un árbol con un vértice distinguido como raíz y tal que cada arista apunta del vértice más cercano a la raíz al más lejano.
- Si en un árbol dirigido existe la arista ij, se dice que i es el padre de j, o que j es un hijo de i.
- Obs: Todo vértice tiene un solo padre, salvo la raíz que no tiene. Se llaman hojas los vértices sin hijos.
- Se define el nivel de un vértice como su distancia a la raíz.



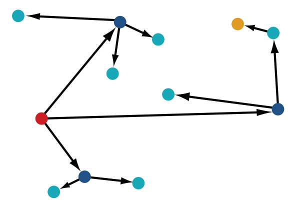
Árboles dirigidos

- Un árbol dirigido es un árbol con un vértice distinguido como raíz y tal que cada arista apunta del vértice más cercano a la raíz al más lejano.
- Si en un árbol dirigido existe la arista ij, se dice que i es el padre de j, o que j es un hijo de i.
- <u>Obs:</u> Todo vértice tiene un solo padre, salvo la raíz que no tiene. Se llaman hojas los vértices sin hijos.
- Se define el nivel de un vértice como su distancia a la raíz.



Árboles dirigidos

- Un árbol dirigido es un árbol con un vértice distinguido como raíz y tal que cada arista apunta del vértice más cercano a la raíz al más lejano.
- Si en un árbol dirigido existe la arista ij, se dice que i es el padre de j, o que j es un hijo de i.
- Obs: Todo vértice tiene un solo padre, salvo la raíz que no tiene. Se llaman hojas los vértices sin hijos.
- Se define el nivel de un vértice como su distancia a la raíz.



Lema 1

Si $m > n - 1$ entonces G tiene un ciclo.

Lema 1

Si $m > n - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n = 1$ o 2, no puede pasar. Si $n = 3$, entonces G es un triángulo y tiene un ciclo.

Lema 1

Si $m > n - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n = 1$ o 2, no puede pasar. Si $n = 3$, entonces G es un triángulo y tiene un ciclo.
Sea G con $n_G > 3$ y $m_G > n_G - 1$. Si todo vértice de G tiene grado al menos 2, entonces G tiene un ciclo (ej. práctica).

Lema 1

Si $m > n - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n = 1$ o 2, no puede pasar. Si $n = 3$, entonces G es un triángulo y tiene un ciclo.
Sea G con $n_G > 3$ y $m_G > n_G - 1$. Si todo vértice de G tiene grado al menos 2, entonces G tiene un ciclo (ej. práctica).
Si no, saco un vértice v con $d(v) \leq 1$. Ahora $G' = G - v$ cumple

$$m_{G'} \geq m_G - 1 > n_G - 2 = n_{G'} - 1$$

Lema 1

Si $m > n - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n = 1$ o 2, no puede pasar. Si $n = 3$, entonces G es un triángulo y tiene un ciclo.
Sea G con $n_G > 3$ y $m_G > n_G - 1$. Si todo vértice de G tiene grado al menos 2, entonces G tiene un ciclo (ej. práctica).
Si no, saco un vértice v con $d(v) \leq 1$. Ahora $G' = G - v$ cumple

$$m_{G'} \geq m_G - 1 > n_G - 2 = n_{G'} - 1$$

Por hipótesis inductiva G' contiene un ciclo, y como G' es un subgrafo de G, es también un ciclo en G. $\square$

Lema 2

Si G es conexo, entonces $m \geq n - 1$.

Lema 2

Si G es conexo, entonces $m \geq n - 1$.

Demo: Por inducción. Si $n = 1$ o 2, se verifica.

Lema 2

Si G es conexo, entonces $m \geq n - 1$.

Demo: Por inducción. Si $n = 1$ o 2, se verifica.
Sea G conexo, $n_G \geq 3$. Si todo vértice de G tiene grado al menos 2, entonces $2m_G \geq 2n_G$, luego $m \geq n - 1$.

Lema 2

Si G es conexo, entonces $m \geq n - 1$.

Demo: Por inducción. Si $n = 1$ o 2, se verifica.
Sea G conexo, $n_G \geq 3$. Si todo vértice de G tiene grado al menos 2, entonces $2m_G \geq 2n_G$, luego $m \geq n - 1$.
Si no, sea v de grado 1 (no puede haber vértices de grado cero por ser G conexo no trivial). Como v no puede ser punto de corte, $G' = G - v$ es conexo.

Lema 2

Si G es conexo, entonces $m \geq n - 1$.

Demo: Por inducción. Si $n = 1$ o 2, se verifica.
Sea G conexo, $n_G \geq 3$. Si todo vértice de G tiene grado al menos 2, entonces $2m_G \geq 2n_G$, luego $m \geq n - 1$.
Si no, sea v de grado 1 (no puede haber vértices de grado cero por ser G conexo no trivial). Como v no puede ser punto de corte, $G' = G - v$ es conexo.
Por hipótesis inductiva $m_{G'} \geq n_{G'} - 1$. Entonces

$$m_G = m_{G'} + 1 \geq n_{G'} = n_G - 1$$

$\square$

Teorema

Son equivalentes:

1. G es un árbol.
2. Todo par de vértices de G está unido por un único camino.
3. G es conexo y $m = n - 1$.
4. G es acíclico y $m = n - 1$.

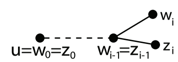
G es un árbol $\Leftrightarrow$ todo par de vértices de G está unido por un único camino

Demo: $\Leftarrow$) Si todo par de vértices de G está unido por un único camino, claramente G es conexo. Además, si hubiera un ciclo, cualquier par de vértices del ciclo estaría unido por al menos dos caminos distintos. Luego G es un árbol.

G es un árbol $\Leftrightarrow$ todo par de vértices de G está unido por un único camino

Demo: $\Leftarrow$) Si todo par de vértices de G está unido por un único camino, claramente G es conexo. Además, si hubiera un ciclo, cualquier par de vértices del ciclo estaría unido por al menos dos caminos distintos. Luego G es un árbol.

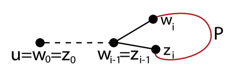
$\Rightarrow$) Si G es un árbol, es conexo, luego todo par de vértices está unido por al menos un camino. Supongamos que u y v están unidos por al menos dos caminos distintos, $P_1 : u = w_0, w_1, \ldots, w_k = v$ y
$P_2 : u = z_0, z_1, \ldots, z_r = v$. Sea i el primer índice tal que $w_i \neq z_i$.
Entonces $i > 0$, y $w_{i-1} = z_{i-1}$.



G es un árbol $\Leftrightarrow$ todo par de vértices de G está unido por un único camino

Demo: $\Leftarrow$) Si todo par de vértices de G está unido por un único camino, claramente G es conexo. Además, si hubiera un ciclo, cualquier par de vértices del ciclo estaría unido por al menos dos caminos distintos. Luego G es un árbol.

$\Rightarrow$) Si G es un árbol, es conexo, luego todo par de vértices está unido por al menos un camino. Supongamos que u y v están unidos por al menos dos caminos distintos, $P_1 : u = w_0, w_1, \ldots, w_k = v$ y
$P_2 : u = z_0, z_1, \ldots, z_r = v$. Sea i el primer índice tal que $w_i \neq z_i$.
Entonces $i > 0$, y $w_{i-1} = z_{i-1}$.



Además, $w_i, \ldots, w_k = v = z_r, \ldots, z_i$ inducen en G un subgrafo conexo.
Sea P un camino mínimo entre w_i y z_i en ese subgrafo inducido.
Entonces $w_{i-1} P w_{i-1}$ es un ciclo en G, absurdo. □

G es un árbol $\Leftrightarrow$ G es conexo y $m = n - 1$

Demo: $\Rightarrow$) Si G es un árbol entonces es conexo y por Lema 2, $n \geq m - 1$. Además, como es acíclico, por Lema 1, $n \leq m - 1$. Luego $n = m - 1$.

G es un árbol $\Leftrightarrow$ G es conexo y $m = n - 1$

Demo: $\Rightarrow$) Si G es un árbol entonces es conexo y por Lema 2, $n \geq m - 1$. Además, como es acíclico, por Lema 1, $n \leq m - 1$. Luego $n = m - 1$.

$\Leftarrow$) G es conexo, probemos por inducción que es un árbol. Si $n = 1$, vale. Supongamos $n > 1$. Por propiedad de la suma de grados, G tiene al menos un vértice v de grado menor o igual que uno. Como es conexo, v tiene grado 1, y entonces no es punto de corte. Luego $G' = G - v$ es conexo y

$$n_{G'} - 1 = n_G - 2 = m_G - 1 = m_{G'}.$$

Por hipótesis inductiva G' es un árbol, y entonces G era un árbol también (v tiene grado 1, no puede pertenecer a un ciclo). $\square$

$$G \text{ es un árbol} \Leftrightarrow G \text{ es acíclico y } m = n - 1$$

Demo: $\Rightarrow$) Si G es un árbol entonces es acíclico y por Lema 1, $m \leq n-1$. Además, como es conexo, por Lema 2, $m \geq n-1$. Luego $m = n-1$.

G es un árbol $\Leftrightarrow$ G es acíclico y $m = n - 1$

Demo: $\Rightarrow$) Si G es un árbol entonces es acíclico y por Lema 1, $m \leq n - 1$. Además, como es conexo, por Lema 2, $m \geq n - 1$. Luego $m = n - 1$.

$\Leftarrow$) G es acíclico. Supongamos que tiene t componentes conexas $G_1, \ldots, G_t$. Cada una de ellas es un árbol, y por la equivalencia anterior, $m_{G_i} = n_{G_i} - 1$. Pero entonces

$$m_G = \sum_{i=1}^{t} m_{G_i} = \sum_{i=1}^{t} (n_{G_i} - 1) = \sum_{i=1}^{t} n_{G_i} - t = n_G - t.$$

Luego $t = 1$, por lo tanto G es conexo. $\square$

Teorema

Todo árbol no trivial tiene al menos dos hojas.

Teorema

Todo árbol no trivial tiene al menos dos hojas.

Demo: Si T es un árbol no trivial, por ser conexo tiene una arista $v_0 w_0$. O v_0 es una hoja, o puedo elegir un vecino v_1 de v_0, tal que $v_1 \neq w_0$. En cada paso, o v_i es una hoja o tiene un vecino distinto de v_{i-1} y también distinto del resto de los vértices del camino, porque T es acíclico. Como los vértices son finitos, hay algún v_k que es una hoja. Con el mismo argumento a partir de w_0, hay algún w_t que es una hoja, y es distinto de todos los v_i. $\square$

Teorema

1. Toda arista de un árbol es un puente.
2. Un vértice de un árbol no trivial es un punto de corte sii no es una hoja.

Teorema

1. Toda arista de un árbol es un puente.
2. Un vértice de un árbol no trivial es un punto de corte sii no es una hoja.

Demo: 1. Sea T un árbol y vw una arista de T. Como en T hay un único camino entre v y w (es v-e-w), no existe camino que una v y w en $T - e$.

Teorema

1. Toda arista de un árbol es un puente.
2. Un vértice de un árbol no trivial es un punto de corte sii no es una hoja.

Demo: 1. Sea T un árbol y vw una arista de T. Como en T hay un único camino entre v y w (es v-e-w), no existe camino que una v y w en $T - e$.
2. En cualquier grafo, una hoja nunca puede ser punto de corte. Sea T un árbol no trivial y v vértice de T que no es hoja. Entonces tiene al menos dos vecinos, z y w. Como en T existe un único camino que une a z y w (es zvw), no existe camino entre z y w en $T - v$. $\square$

Teorema

Un grafo es conexo sii admite un árbol generador.

Teorema

Un grafo es conexo sii admite un árbol generador.

Idea de demo: $\Leftarrow$) Sea T un a.g. de G. Sean v, w vértices de G. En T existe un camino de v a w. Como T es subgrafo de G, en particular es un camino en G de v a w. Por lo tanto G es conexo.

Teorema

Un grafo es conexo sii admite un árbol generador.

Idea de demo: $\Leftarrow$) Sea T un a.g. de G. Sean v, w vértices de G. En T existe un camino de v a w. Como T es subgrafo de G, en particular es un camino en G de v a w. Por lo tanto G es conexo.

$\Leftarrow$) Por inducción. Si G es trivial, G es un a.g. de si mismo. Si no, sea v un vértice de G. Por hipótesis inductiva, cada componente conexa G_i de $G - v$ tiene un a.g. T_i. Como G era conexo, v tiene un vecino v_i en cada G_i. Sea T el grafo obtenido agregando a la unión de los T_i el vértice v y las aristas $v_i v$. T es un a.g. de G: es fácil ver que es conexo y que $m_T = n_T - 1$, sabiendo que los T_i son conexos y $m_{T_i} = n_{T_i} - 1$. $\square$

Corolario

Todo grafo conexo no trivial tiene al menos dos vértices tales que al sacar alguno de ellos, sigue siendo conexo.

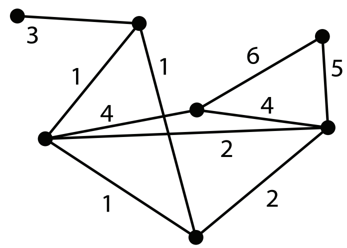
Corolario

Todo grafo conexo no trivial tiene al menos dos vértices tales que al sacar alguno de ellos, sigue siendo conexo.

Idea de demo: Si G es conexo, tiene un árbol generador T. Como G es no trivial, T también, y por lo tanto tiene al menos dos hojas v, w. Como $T - v$ y $T - w$ siguen siendo árboles, son árboles generadores de $G - v$ y $G - w$, respectivamente. Luego $G - v$ y $G - w$ son conexos. $\square$

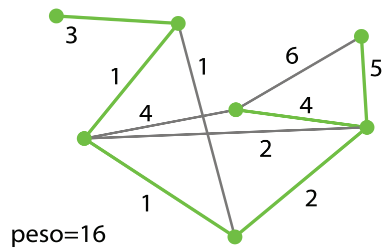
Árbol generador mínimo

- Un grafo pesado es un grafo tal que sus aristas tienen asociado un peso.
- El peso de un subgrafo es la suma de los pesos de sus aristas.
- Un árbol generador mínimo en un grafo pesado es un árbol generador de peso mínimo.



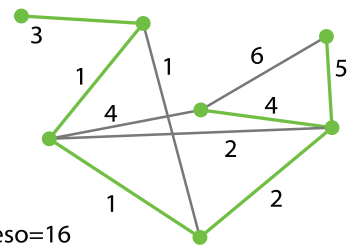
Árbol generador mínimo

- Un grafo pesado es un grafo tal que sus aristas tienen asociado un peso.
- El peso de un subgrafo es la suma de los pesos de sus aristas.
- Un árbol generador mínimo en un grafo pesado es un árbol generador de peso mínimo.



Árbol generador mínimo

- Un grafo pesado es un grafo tal que sus aristas tienen asociado un peso.
- El peso de un subgrafo es la suma de los pesos de sus aristas.
- Un árbol generador mínimo en un grafo pesado es un árbol generador de peso mínimo.



peso=16

Ejemplo de aplicación

Tengo que construir caminos entre ciertos pares de ciudades, de modo que todo el país me quede conectado. ¿Cómo puedo hacerlo minimizando la longitud total de camino construido?

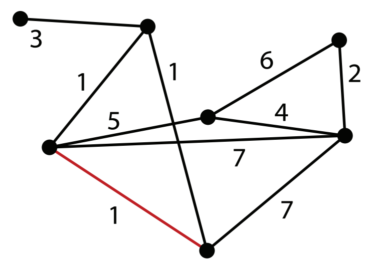
Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n - 1$ aristas.

Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

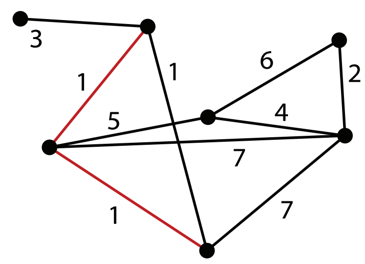
Ej:



Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

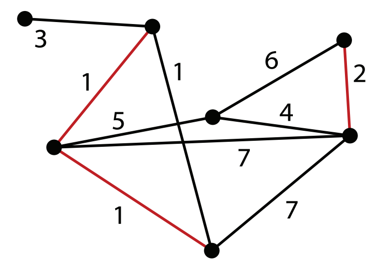
Ej:



Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

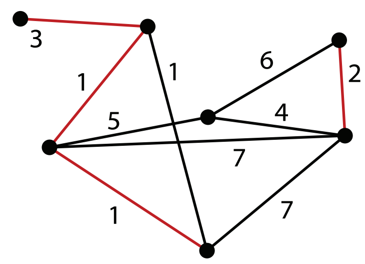
Ej:



Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

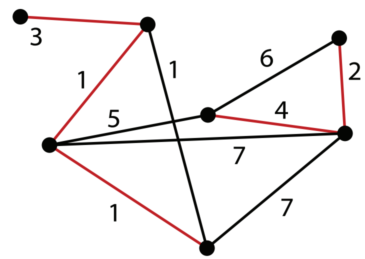
Ej:



Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

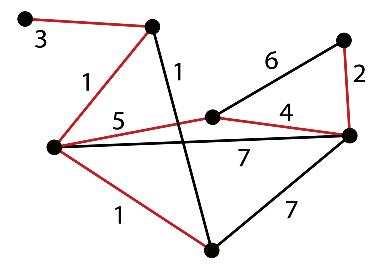
Ej:



Algoritmo de Kruskal para AGM

Partir de un subgrafo generador cuyo conjunto de aristas es vacío, y en cada paso agregar una arista de peso mínimo que no forme ciclos con las demás aristas del conjunto, hasta haber agregado $n-1$ aristas.

Ej:



Demostración de que Kruskal construye un AGM

Para ver que el algoritmo construye un árbol generador, como en cada paso el subgrafo B elegido hasta el momento es generador y acíclico, basta ver que el algoritmo termina con $m_B = n_G - 1$. Si $m_B < n_G - 1$, B es no conexo. Sea B_1 una componente conexa de B. Como G es conexo, va a existir alguna arista con un extremo en B_1 y el otro en $V(G) - B_1$, que por lo tanto no forma ciclo con las demás aristas de B. Entonces, si $m_B < n_G - 1$, el algoritmo puede realizar un paso más.

Sea G un grafo, T_K el árbol generado por el algoritmo de Kruskal y $\{e_1, e_2, \ldots, e_{n-1}\}$ la secuencia de aristas de T_K en el orden en que fueron elegidas por el algoritmo de Kruskal. Para cada árbol generador T de G definimos $p(T)$ como el máximo $k \leq n$ tal que $\forall j < k$, $e_j \in T$.

Demostración de que Kruskal construye un AGM

Para ver que el algoritmo construye un árbol generador, como en cada paso el subgrafo B elegido hasta el momento es generador y acíclico, basta ver que el algoritmo termina con $m_B = n_G - 1$. Si $m_B < n_G - 1$, B es no conexo. Sea B_1 una componente conexa de B. Como G es conexo, va a existir alguna arista con un extremo en B_1 y el otro en $V(G) - B_1$, que por lo tanto no forma ciclo con las demás aristas de B. Entonces, si $m_B < n_G - 1$, el algoritmo puede realizar un paso más.

Sea G un grafo, T_K el árbol generado por el algoritmo de Kruskal y $\{e_1, e_2, \dots, e_{n-1}\}$ la secuencia de aristas de T_K en el orden en que fueron elegidas por el algoritmo de Kruskal. Para cada árbol generador T de G definimos $p(T)$ como el máximo $k \leq n$ tal que $\forall j < k$, $e_j \in T$.

Demostración de que Kruskal construye un AGM

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con T_K, con lo cual T_K resulta ser mínimo. Si T_K no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Como T_K es acíclico, hay alguna arista e en C tal que $e \notin T_K$. Como $e_1, \ldots, e_{p(T)-1} \in T$ y T es acíclico, e no forma ciclos con $e_1, \ldots, e_{p(T)-1}$. Por la forma en que fue elegida $e_{p(T)}$ por el algoritmo de Kruskal, $peso(e_{p(T)}) \leq peso(e)$.

Pero entonces $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.

Luego T_K es un árbol generador mínimo. $\square$

Demostración de que Kruskal construye un AGM

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con T_K, con lo cual T_K resulta ser mínimo. Si T_K no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Como T_K es acíclico, hay alguna arista e en C tal que $e \notin T_K$. Como $e_1, \dots, e_{p(T)-1} \in T$ y T es acíclico, e no forma ciclos con $e_1, \dots, e_{p(T)-1}$. Por la forma en que fue elegida $e_{p(T)}$ por el algoritmo de Kruskal, $peso(e_{p(T)}) \leq peso(e)$.

Pero entonces $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.

Luego T_K es un árbol generador mínimo. $\square$

Demostración de que Kruskal construye un AGM

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con T_K, con lo cual T_K resulta ser mínimo. Si T_K no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Como T_K es acíclico, hay alguna arista e en C tal que $e \notin T_K$. Como $e_1, \ldots, e_{p(T)-1} \in T$ y T es acíclico, e no forma ciclos con $e_1, \ldots, e_{p(T)-1}$. Por la forma en que fue elegida $e_{p(T)}$ por el algoritmo de Kruskal, $peso(e_{p(T)}) \leq peso(e)$.

Pero entonces $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.

Luego T_K es un árbol generador mínimo. $\square$

Demostración de que Kruskal construye un AGM

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con T_K, con lo cual T_K resulta ser mínimo. Si T_K no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Como T_K es acíclico, hay alguna arista e en C tal que $e \notin T_K$. Como $e_1, \dots, e_{p(T)-1} \in T$ y T es acíclico, e no forma ciclos con $e_1, \dots, e_{p(T)-1}$. Por la forma en que fue elegida $e_{p(T)}$ por el algoritmo de Kruskal, $peso(e_{p(T)}) \leq peso(e)$.

Pero entonces $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.

Luego T_K es un árbol generador mínimo. $\square$

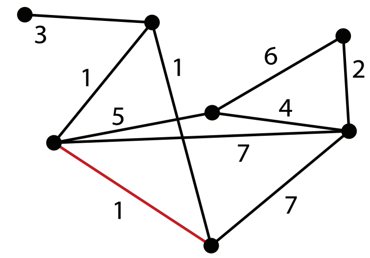
Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

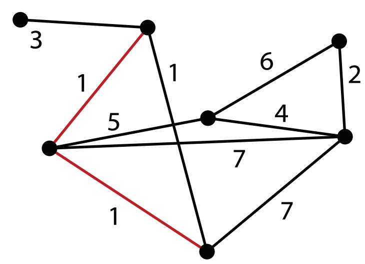
Ej:



Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

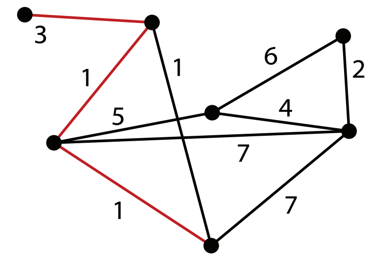
Ej:



Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

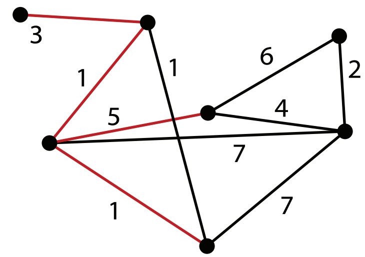
Ej:



Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

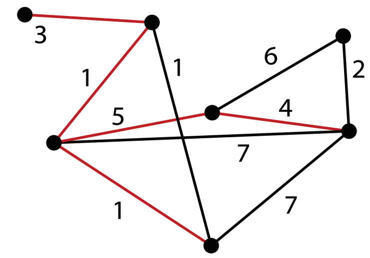
Ej:



Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

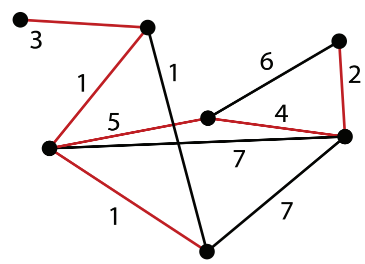
Ej:



Algoritmo de Prim para AGM

Partir de un conjunto de aristas $A = \{e\}$ y un conjunto de vértices $W = \{v, w\}$, donde e es una arista de peso mínimo en G y v y w son sus extremos. En cada paso, agregar a A una arista f de peso mínimo con un extremo en W y el otro en $V(G) - W$. Agregar a W el extremo de f que no estaba en W, hasta que $W = V(G)$.

Ej:



Demostración de que Prim construye un AGM

Para ver que construye un árbol generador, se puede ver que en cada paso del algoritmo, el subgrafo elegido hasta el momento es conexo y con $m = n - 1$. Finalmente, como el grafo es conexo, mientras $W \neq V(G)$ va a existir alguna arista de W a $V(G) - W$ con lo cual el algoritmo termina construyendo un árbol generador del grafo.

Sea G un grafo, P el árbol generado por el algoritmo de Prim y $\{e_1, e_2, \ldots, e_{n-1}\}$ la secuencia de aristas de P en el orden en que fueron elegidas por el algoritmo de Prim. Para cada árbol generador T de G definimos $p(T)$ como el máximo $k \leq n$ tal que $\forall j \leq k$, $e_j \in T$.

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con P, con lo cual P resulta ser mínimo. Si P no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Demostración de que Prim construye un AGM

Para ver que construye un árbol generador, se puede ver que en cada paso del algoritmo, el subgrafo elegido hasta el momento es conexo y con $m = n - 1$. Finalmente, como el grafo es conexo, mientras $W \neq V(G)$ va a existir alguna arista de W a $V(G) - W$ con lo cual el algoritmo termina construyendo un árbol generador del grafo.

Sea G un grafo, P el árbol generado por el algoritmo de Prim y $\{e_1, e_2, \dots, e_{n-1}\}$ la secuencia de aristas de P en el orden en que fueron elegidas por el algoritmo de Prim. Para cada árbol generador T de G definimos $p(T)$ como el máximo $k \leq n$ tal que $\forall j \leq k$, $e_j \in T$.

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con P, con lo cual P resulta ser mínimo. Si P no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Demostración de que Prim construye un AGM

Para ver que construye un árbol generador, se puede ver que en cada paso del algoritmo, el subgrafo elegido hasta el momento es conexo y con $m = n - 1$. Finalmente, como el grafo es conexo, mientras $W \neq V(G)$ va a existir alguna arista de W a $V(G) - W$ con lo cual el algoritmo termina construyendo un árbol generador del grafo.

Sea G un grafo, P el árbol generado por el algoritmo de Prim y $\{e_1, e_2, \ldots, e_{n-1}\}$ la secuencia de aristas de P en el orden en que fueron elegidas por el algoritmo de Prim. Para cada árbol generador T de G definimos $p(T)$ como el máximo $k \leq n$ tal que $\forall j \leq k$, $e_j \in T$.

Ahora, sea T un AGM que maximiza p. Si $p(T) = n$, entonces T coincide con P, con lo cual P resulta ser mínimo. Si P no es mínimo, entonces $p(T) < n$ y $e_{p(T)} \notin T$. Como T es conexo, en T hay un camino C que une los extremos de $e_{p(T)}$.

Demostración de que Prim construye un AGM

- Si $p(T) = 1$, como e_1 es de peso mínimo, $peso(e_1) \leq peso(e)$ $\forall e \in C$. Luego $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.
- Si $p(T) > 1$, sea V_1 el conjunto de extremos de las aristas $e_1, \ldots, e_{p(T)-1}$ y $V_2 = V - V_1$. Por la forma de elegir las aristas en Prim, $e_{p(T)}$ es de peso mínimo entre las que tienen un extremo en V_1 y otro en V_2. El camino C va de un vértice de V_1 a un vértice de V_2, con lo cual, existe $e \in C$ con un extremo en V_1 y otro en V_2 (sus vértices se pueden partir entre los de V_1 y los de V_2, ambos conjuntos son no vacíos y C es conexo). Entonces $peso(e_{p(T)}) \leq peso(e)$ y $T - e \cup \{e_{p(T)}\}$ es un árbol generador de peso menor o igual a T y p mayor a $p(T)$ (e no es ninguna de las e_i con $i < p(T)$ porque esas tienen ambos extremos en V_1, por definición de V_1), absurdo.

Luego P es un árbol generador mínimo. $\square$

Demostración de que Prim construye un AGM

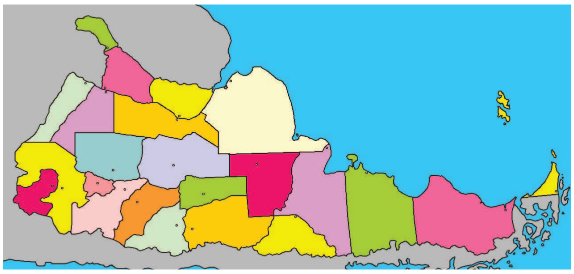
- Si $p(T) = 1$, como e_1 es de peso mínimo, $peso(e_1) \leq peso(e)$ $\forall e \in C$. Luego $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.
- Si $p(T) > 1$, sea V_1 el conjunto de extremos de las aristas $e_1, \ldots, e_{p(T)-1}$ y $V_2 = V - V_1$. Por la forma de elegir las aristas en Prim, $e_{p(T)}$ es de peso mínimo entre las que tienen un extremo en V_1 y otro en V_2. El camino C va de un vértice de V_1 a un vértice de V_2, con lo cual, existe $e \in C$ con un extremo en V_1 y otro en V_2 (sus vértices se pueden partir entre los de V_1 y los de V_2, ambos conjuntos son no vacíos y C es conexo). Entonces $peso(e_{p(T)}) \leq peso(e)$ y $T - e \cup \{e_{p(T)}\}$ es un árbol generador de peso menor o igual a T y p mayor a $p(T)$ (e no es ninguna de las e_i con $i < p(T)$ porque esas tienen ambos extremos en V_1, por definición de V_1), absurdo.

Luego P es un árbol generador mínimo. $\square$

Demostración de que Prim construye un AGM

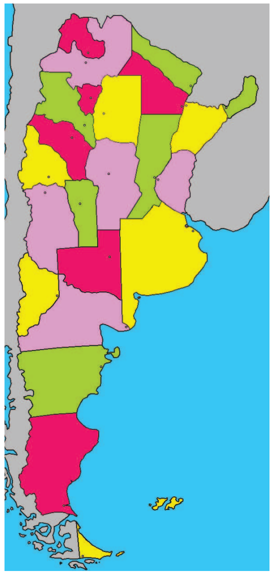
- Si $p(T) = 1$, como e_1 es de peso mínimo, $peso(e_1) \leq peso(e)$ $\forall e \in C$. Luego $T' = T - e \cup \{e_{p(T)}\}$ es un árbol generador de G de peso menor o igual a T y $p(T') > p(T)$, absurdo.
- Si $p(T) > 1$, sea V_1 el conjunto de extremos de las aristas $e_1, \ldots, e_{p(T)-1}$ y $V_2 = V - V_1$. Por la forma de elegir las aristas en Prim, $e_{p(T)}$ es de peso mínimo entre las que tienen un extremo en V_1 y otro en V_2. El camino C va de un vértice de V_1 a un vértice de V_2, con lo cual, existe $e \in C$ con un extremo en V_1 y otro en V_2 (sus vértices se pueden partir entre los de V_1 y los de V_2, ambos conjuntos son no vacíos y C es conexo). Entonces $peso(e_{p(T)}) \leq peso(e)$ y $T - e \cup \{e_{p(T)}\}$ es un árbol generador de peso menor o igual a T y p mayor a $p(T)$ (e no es ninguna de las e_i con $i < p(T)$ porque esas tienen ambos extremos en V_1, por definición de V_1), absurdo.

Luego P es un árbol generador mínimo. $\square$



Pintando mapas...

Pintando mapas...



La Conjetura de los Cuatro Colores

La Conjetura de los Cuatro Colores fue enunciada en el siglo XIX:

Todo mapa puede ser coloreado usando a lo sumo cuatro colores de manera tal que regiones limítrofes (i.e. que comparten una frontera de dimensión uno, no sólo un punto) reciban colores diferentes.

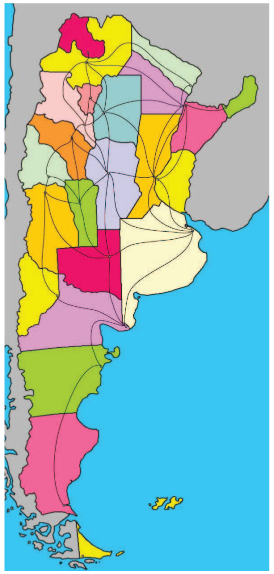
En términos de grafos...

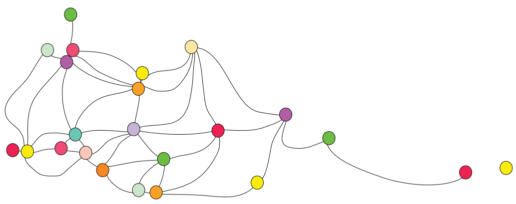
A partir de cualquier mapa podemos construir un grafo donde las regiones se representan por vértices y dos vértices son adyacentes si y sólo si las regiones correspondientes son limítrofes.

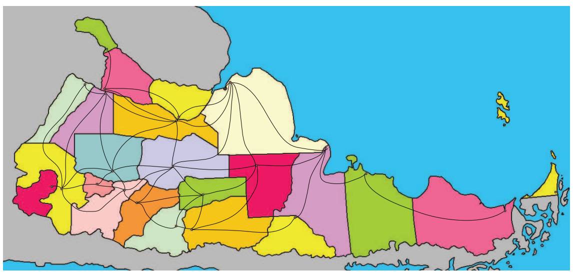
El grafo resultante es planar, es decir, se puede dibujar en el plano sin aristas que se crucen.

Entonces, la Conjetura de los Cuatro Colores dice que los vértices de un grafo planar pueden ser coloreados usando a lo sumo 4 colores y de manera tal que no haya dos vértices adyacentes que reciban el mismo color.

Ejemplo...







Ejemplo...

Historia

Aparentemente, la Conjetura de los Cuatro Colores fue formulada por primera vez por Francis Guthrie. Él era un estudiante del University College de Londres y fue alumno de Augusts De Morgan.

Después de graduarse en Londres estudió derecho, pero algunos años después su hermano Frederick Guthrie comenzó a estudiar con De Morgan. Francis le mostró a su hermano algunos resultados que había estado intentando probar sobre coloreo de mapas y le pidió que le pregunte a De Morgan sobre ese problema.

Guthrie

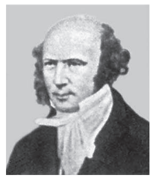
De Morgan

De Morgan no pudo darle una respuesta pero ese mismo día (el 23 de octubre de 1852) le escribió una carta a Sir William Hamilton en Dublin:

A student of mine asked me today to give him a reason for a fact which I did not know was a fact - and do not yet. He says that if a figure be anyhow divided and the compartments differently colored so that figures with any portion of common boundary line are differently colored - *four colors may be wanted, but not more* - the following is the case in which four colors are wanted. Query cannot a necessity for five or more be invented. ... If you retort with some very simple case which makes me out a stupid animal, I think I must do as the Sphynx did...

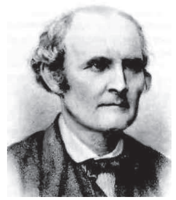
Hamilton le respondió el 26 de octubre de 1852 (mostrando tanto su eficiencia como la del correo):

I am not likely to attempt your quaternion of colors very soon.



Hamilton

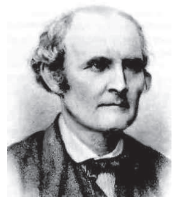
La primera referencia publicada se encuentra en el paper de Arthur Cayley, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.



Cayley

La primera referencia publicada se encuentra en el paper de Arthur Cayley, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.

El 17 de julio de 1879 Alfred Bray Kempe anunció en Nature que tenía una demostración de la Conjetura de los Cuatro Colores.



Cayley

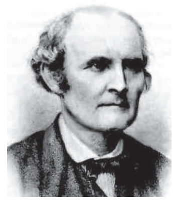
Kempes

La primera referencia publicada se encuentra en el paper de Arthur Cayley, *On the colorings of maps*, Proc. Royal Geographical Society 1, 259–261, 1879.

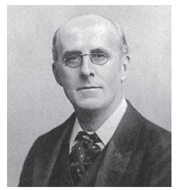
El 17 de julio de 1879 Alfred Bray Kempe anunció en Nature que tenía una demostración de la Conjetura de los Cuatro Colores.

Kempe era un abogado de Londres que había estudiado matemática con Cayley en Cambridge y dedicado algo de su tiempo a la matemática durante su vida.

Por sugerencia de Cayley, Kempe envió su Teorema al American Journal of Mathematics donde fue publicado a fines de 1879.



Cayley



Kempe

Idea de la demostración de Kempe

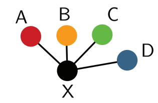
Kempe usó un argumento inductivo-constructivo:

Si tenemos un mapa en el cual cada región es coloreada con rojo, verde, azul o amarillo excepto una, digamos X. Si X no está rodeada por regiones de todos los colores, me queda un color para asignarle a X. Supongamos entonces que X tiene regiones limítrofes de los 4 colores.

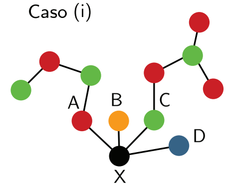
Si X está rodeada por regiones A, B, C, D en orden, coloreadas rojo, amarillo, verde y azul entonces hay dos casos a considerar.

(i) No existe una cadena de regiones adyacentes uniendo A y C, coloreadas alternadamente de rojo y verde.

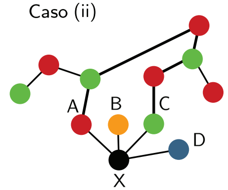
(ii) Existe una cadena de regiones adyacentes uniendo A y C, coloreadas alternadamente de rojo y verde.



Caso (i)



Caso (ii)

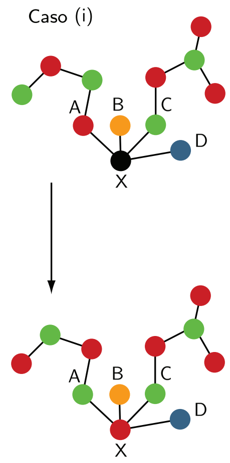


Casos:

(i) No existe una cadena roja/verde uniendo A y C.

(ii) Existe una cadena roja/verde uniendo A y C.

Si vale (i), entonces cambiamos A a verde, y luego intercambiamos los colores rojo/verde en la componente bicromática rojo/verde que contiene a A. Como C no está en la componente, permanece verde y ahora no hay más regiones rojas adyacentes a X. Coloreamos X con rojo.



Casos:

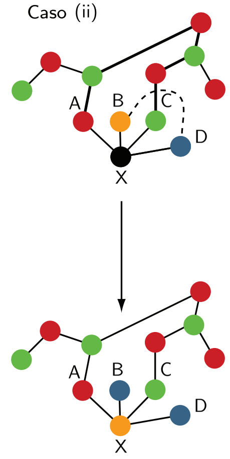
(i) No existe una cadena roja/verde uniendo A y C.

(ii) Existe una cadena roja/verde uniendo A y C.

Si vale (i), entonces cambiamos A a verde, y luego intercambiamos los colores rojo/verde en la componente bicromática rojo/verde que contiene a A. Como C no está en la componente, permanece verde y ahora no hay más regiones rojas adyacentes a X. Coloreamos X con rojo.

Si vale (ii), entonces no puede haber una cadena amarilla/azul de regiones adyacentes uniendo B y D. [No podría cruzar la cadena verde/roja.] Entonces la propiedad (i) vale para B y D y cambiamos los colores como antes.

Caso (ii)



El Teorema de los Cuatro Colores volvió a ser Conjetura de los Cuatro Colores en 1890.

Percy John Heawood, un docente en Durham England, publicó un paper llamado *Map coloring theorem*. Ahi decía que su objetivo era "*...más destructivo que constructivo, ya que voy a mostrar un defecto en la aparentemente reconocida prueba...*".

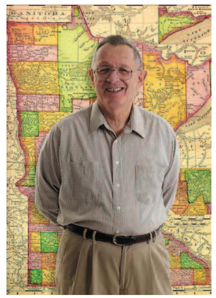
Aunque Heawood mostró que la prueba de Kempe era errónea, él probó en ese paper que todo mapa puede ser coloreado usando a lo sumo 5 colores.

Heawood

Primera prueba

Finalmente, en 1976 (si, 100 años después...) la Conjetura de los Cuatro Colores fue probada por Kenneth Appel y Wolfgang Haken en la Universidad de Illinois. John Koch colaboró con partes del algoritmo.

- K. Appel and W. Haken, *Every planar map is four colorable. Part I. Discharging*, Illinois J. Math. 21 (1977), 429–490.
- K. Appel, W. Haken and J. Koch, *Every planar map is four colorable. Part II. Reducibility*, Illinois J. Math. 21 (1977), 491–567.



Appel

Idea de la demostración

Para demostrar la inexistencia de un contraejemplo (y en particular de un contraejemplo minimal), Appel y Haken redujeron mediante argumentos teóricos el problema de los infinitos mapas posibles a 1,936 configuraciones que debieron ser chequeadas una a una por computadora, con un algoritmo que corrió cientos de horas.

La parte teórica a su vez incluye 500 páginas de contra-contra-ejemplos escritos a mano (esos coloreos fueron verificados por el hijo de Haken!).

Pero muchos científicos cuestionaron la demostración de Appel-Haken por dos motivos:

- Parte de la prueba usaba una computadora, y no podía ser verificada a mano.
- Aún la parte supuestamente chequeable a mano es extremadamente engorrosa, y nadie la verificó por completo.

Segunda prueba

Hace diez años apareció otra prueba:

- N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, *The four color theorem*, J. Combin. Theory Ser. B. 70 (1997), 2–44.
- N. Robertson, D. P. Sanders, P. D. Seymour and R. Thomas, *A new proof of the four color theorem*, Electron. Res. Announc. Amer. Math. Soc. 2 (1996), 17–25 (electronic).


Robertson


Sanders


Seymour


Thomas

Esquema de la demostración

La idea básica de la prueba es la misma que la de Appel y Haken. Los autores exhibieron un conjunto de 633 configuraciones reducibles, es decir, configuraciones que no pueden aparecer en un contraejemplo minimal. Usaron también un resultado conocido desde 1913 sobre la estructura que debería tener un tal contraejemplo.

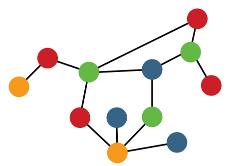
En la segunda parte de la demostración, probaron que todo grafo con dicha estructura contiene alguna de las 633 configuraciones reducibles, y por lo tanto no existe tal contraejemplo.
La primera parte de la prueba requiere de una computadora, pero la segunda puede ser chequeada a mano en pocos meses o, usando una computadora, en unos 20 minutos.

En diciembre de 2004 en una reunión científica en Francia, un grupo de gente de Microsoft Research en Inglaterra e INRIA en Francia anunciaron la verificación de la demostración de Robertson et al. formulando el problema en el lenguaje Coq y confirmando la validez de cada uno de sus pasos (Devlin 2005, Knight 2005).

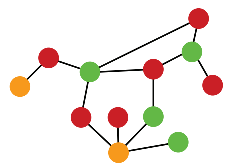
De todas formas, no se conoce hasta el momento una demostración del Teorema de los Cuatro Colores al estilo tradicional (completamente verificable por humanos con lápiz y papel).

Coloreo de grafos

- Un k-coloreo de un grafo G es una asignación de colores a sus vértices de manera que no se usan más de k colores y no hay dos vértices adyacentes que reciban el mismo color.
- Un grafo se dice k-coloreable si admite un k-coloreo.



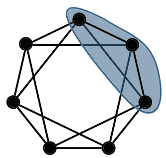
4-coloreo



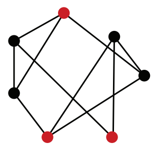
3-coloreo

Número cromático

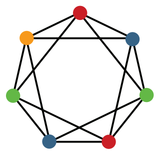
- Una clique en un grafo es un subgrafo completo maximal. Se denota por $\omega(G)$ el tamaño de una clique máxima.
- Un conjunto independiente es un conjunto de vértices no adyacentes dos a dos. Se denota por $\alpha(G)$ el tamaño de un conjunto independiente máximo.
- El número cromático de un grafo G es el menor número k tal que G es k-coloreable, y se denota por $\chi(G)$.



$\omega = 3$



$\alpha = 3$



$\chi = 4$

Aplicaciones

El problema de coloreo de grafos y sus variantes tienen muchas aplicaciones, entre ellas problemas de scheduling, asignación de frecuencias en radios y teléfonos celulares, etc.

Ejemplo: Planificación de exámenes

Cada alumno tiene que rendir un examen en cada una de las materias que está cursando. Sea X el conjunto de materias e Y el conjunto de estudiantes. Como el examen es escrito, es conveniente que todos los alumnos lo rindan a la vez. Por resolución del CD, los alumnos tienen derecho a no rendir dos exámenes el mismo día. ¿Cuál es la mínima cantidad de días de examen necesarios?

Ejemplo: Planificación de exámenes

Modelemos este problema como un problema de coloreo de grafos.

- Vértices: materias.
- Colores: días.
- Aristas: Dos vértices son adyacentes si sus correspondientes materias tienen alumnos en común.
- Correspondencia con el modelo: Dos materias [vértices] no pueden tomar examen el mismo día [usar el mismo color] si hay alumnos que cursan ambas [son adyacentes].

Algunas propiedades de $\chi(G)$, $\omega(G)$ y $\alpha(G)$

Lema

Sea G un grafo de n vértices y $\overline{G}$ su complemento. Entonces:

1. $\alpha(G) = \omega(\overline{G})$
2. $\chi(G) \geq \omega(G)$
3. $\chi(G) \leq \Delta(G) + 1$
4. $\chi(G)\alpha(G) \geq n$
5. $\chi(G) + \alpha(G) \leq n + 1$
6. $\chi(G) + \chi(\overline{G}) \leq n + 1$

Demostraciones:

1. $\alpha(G) = \omega(\overline{G})$

   Por definición, ya que los conjuntos independientes maximales de G son cliques de $\overline{G}$. $\square$

2. $\chi(G) \geq \omega(G)$

   Los vértices de una clique tienen que usar colores distintos, ya que son adyacentes dos a dos. $\square$

3. $\chi(G) \leq \Delta(G) + 1$

   Pintando secuencialmente los vértices con el mínimo color no utilizado por un vecino ya pintado, obtenemos un coloreo válido (no siempre óptimo) con a lo sumo $\Delta(G) + 1$ colores. $\square$

Ej: El coloreo obtenido usa 4 colores, $\Delta(G) + 1 = 6$ y $\chi(G) = 3$.

Demostraciones:

1. $\alpha(G) = \omega(\overline{G})$

   Por definición, ya que los conjuntos independientes maximales de G son cliques de $\overline{G}$. $\square$

2. $\chi(G) \geq \omega(G)$

   Los vértices de una clique tienen que usar colores distintos, ya que son adyacentes dos a dos. $\square$

3. $\chi(G) \leq \Delta(G) + 1$

   Pintando secuencialmente los vértices con el mínimo color no utilizado por un vecino ya pintado, obtenemos un coloreo válido (no siempre óptimo) con a lo sumo $\Delta(G) + 1$ colores. $\square$

Ej: El coloreo obtenido usa 4 colores, $\Delta(G) + 1 = 6$ y $\chi(G) = 3$.

Demostraciones:

1. $\alpha(G) = \omega(\overline{G})$

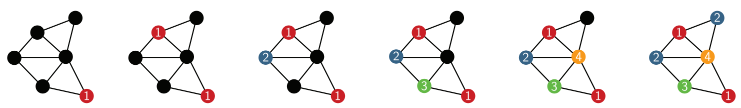
   Por definición, ya que los conjuntos independientes maximales de G son cliques de $\overline{G}$. $\square$

2. $\chi(G) \geq \omega(G)$

   Los vértices de una clique tienen que usar colores distintos, ya que son adyacentes dos a dos. $\square$

3. $\chi(G) \leq \Delta(G) + 1$

   Pintando secuencialmente los vértices con el mínimo color no utilizado por un vecino ya pintado, obtenemos un coloreo válido (no siempre óptimo) con a lo sumo $\Delta(G) + 1$ colores. $\square$



Ej: El coloreo obtenido usa 4 colores, $\Delta(G) + 1 = 6$ y $\chi(G) = 3$.

Demostraciones:

4. $\chi(G)\alpha(G) \geq n$

   Tomemos un coloreo óptimo de G, y llamemos V_i al conjunto de vértices coloreados con i, para $i = 1, \ldots, \chi(G)$. Como cada V_i es un conjunto independiente, $|V_i| \leq \alpha(G)$.
   Entonces $n = \sum_{i=1}^{\chi(G)} |V_i| \leq \chi(G)\alpha(G)$. □

5. $\chi(G) + \alpha(G) \leq n + 1$

   Sea W un conjunto independiente máximo de G. Si pintamos los vértices de W de un color y los de $V - W$ de otros colores diferentes entre si, obtenemos un coloreo válido de G con $1 + n - \alpha(G)$ colores. Luego $\chi(G) \leq n + 1 - \alpha(G)$. □

Ej: $\alpha(G) = 3$, el coloreo obtenido usa $n + 1 - \alpha(G) = 5$ colores, y $\chi(G) = 3$.

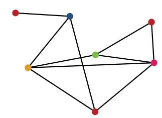
Demostraciones:

4. $\chi(G)\alpha(G) \geq n$

   Tomemos un coloreo óptimo de G, y llamemos V_i al conjunto de vértices coloreados con i, para $i = 1, \ldots, \chi(G)$. Como cada V_i es un conjunto independiente, $|V_i| \leq \alpha(G)$.
   Entonces $n = \sum_{i=1}^{\chi(G)} |V_i| \leq \chi(G)\alpha(G)$. □

5. $\chi(G) + \alpha(G) \leq n + 1$

   Sea W un conjunto independiente máximo de G. Si pintamos los vértices de W de un color y los de $V - W$ de otros colores diferentes entre si, obtenemos un coloreo válido de G con $1 + n - \alpha(G)$ colores. Luego $\chi(G) \leq n + 1 - \alpha(G)$. □



Ej: $\alpha(G) = 3$, el coloreo obtenido usa $n + 1 - \alpha(G) = 5$ colores, y $\chi(G) = 3$.

Demostraciones:

6. $\chi(G)+\chi(\overline{G}) \le n+1$

   Por inducción. Si G es trivial, $\chi(G)+\chi(\overline{G}) = 1+1 = n+1$.
   Si $n \ge 2$, sea $v \in V(G)$. Tenemos dos casos para $G-v$:

   (1) $\chi(G-v) = \chi(G)$
   (2) $\chi(G-v) = \chi(G)-1$.

   Y dos casos para $\overline{G-v} = \overline{G}-v$:

   (a) $\chi(\overline{G-v}) = \chi(\overline{G})$
   (b) $\chi(\overline{G-v}) = \chi(\overline{G})-1$.

   Por hipótesis inductiva, $\chi(G-v)+\chi(\overline{G-v}) \le (n-1)+1$, luego la propiedad para G vale en los casos 1a, 1b y 2a. Supongamos que se da el caso 2b. Si $d_G(v) < \chi(G-v)$ entonces $\chi(G)=\chi(G-v)$. Luego $d_G(v) \ge \chi(G-v) = \chi(G)-1$. Análogamente, $d_{\overline{G}}(v) \ge \chi(\overline{G})-1$.
   Por lo tanto, $n-1 = d_G(v)+d_{\overline{G}}(v) \ge \chi(G)-1+\chi(\overline{G})-1$, entonces $\chi(G)+\chi(\overline{G}) \le n+1$. □

Teorema de Brooks

Teorema de Brooks (1941)

Sea G un grafo conexo. Entonces G es $\Delta(G)$-coloreable, salvo que:

1. $\Delta(G) \neq 2$, y G es un completo de tamaño $\Delta(G) + 1$, or
2. $\Delta(G) = 2$, y G es un ciclo impar.

Algoritmo para coloreo de grafos

No se conoce un algoritmo polinomial para encontrar $\chi(G)$.
Veamos el siguiente algoritmo (conexión-contracción):

- Consideremos un grafo G con dos vértices no adyacentes a y b. La conexión G_1 se obtiene agregando la arista ab. La contracción G_2 se obtiene contrayendo $\{a, b\}$ en un solo vértice $c(a, b)$ que es vecino de $N(a) \cup N(b)$.
- Un coloreo de G en el cual a y b usan distintos colores es un coloreo de G_1, y viceversa. Un coloreo de G en el cual a y b usan el mismo color da lugar a un coloreo de G_2, y viceversa.
- Entonces, $\chi(G) = \text{mín}\{\chi(G_1), \chi(G_2)\}$.
- Si repetimos las operaciones en cada grafo generado hasta que los grafos resultantes sean completos, $\chi(G)$ es el tamaño del menor completo obtenido.

Algoritmo para coloreo de grafos

No se conoce un algoritmo polinomial para encontrar $\chi(G)$.
Veamos el siguiente algoritmo (conexión-contracción):

- Consideremos un grafo G con dos vértices no adyacentes a y b. La conexión G_1 se obtiene agregando la arista ab. La contracción G_2 se obtiene contrayendo $\{a, b\}$ en un solo vértice $c(a, b)$ que es vecino de $N(a) \cup N(b)$.
- Un coloreo de G en el cual a y b usan distintos colores es un coloreo de G_1, y viceversa. Un coloreo de G en el cual a y b usan el mismo color da lugar a un coloreo de G_2, y viceversa.
- Entonces, $\chi(G) = \text{mín}\{\chi(G_1), \chi(G_2)\}$.
- Si repetimos las operaciones en cada grafo generado hasta que los grafos resultantes sean completos, $\chi(G)$ es el tamaño del menor completo obtenido.

Algoritmo para coloreo de grafos

No se conoce un algoritmo polinomial para encontrar $\chi(G)$.
Veamos el siguiente algoritmo (conexión-contracción):

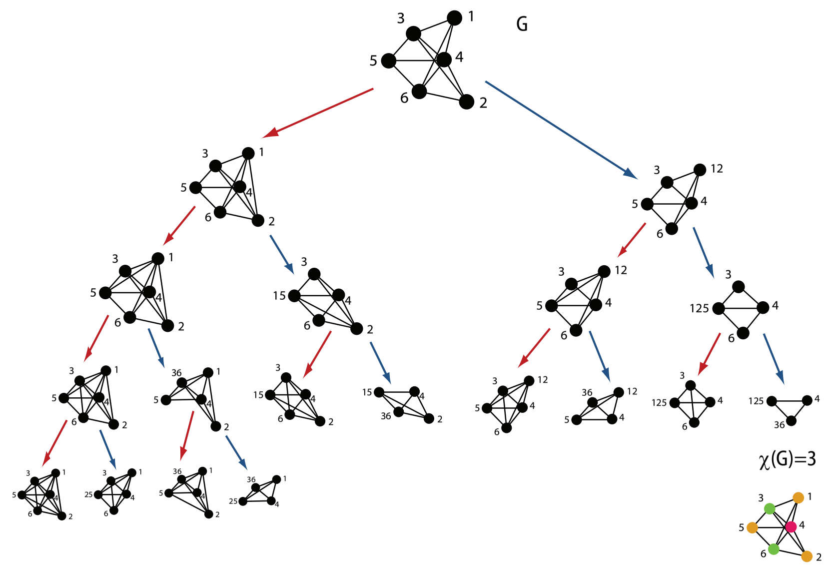
- Consideremos un grafo G con dos vértices no adyacentes a y b. La conexión G_1 se obtiene agregando la arista ab. La contracción G_2 se obtiene contrayendo $\{a, b\}$ en un solo vértice $c(a, b)$ que es vecino de $N(a) \cup N(b)$.
- Un coloreo de G en el cual a y b usan distintos colores es un coloreo de G_1, y viceversa. Un coloreo de G en el cual a y b usan el mismo color da lugar a un coloreo de G_2, y viceversa.
- Entonces, $\chi(G) = \text{mín}\{\chi(G_1), \chi(G_2)\}$.
- Si repetimos las operaciones en cada grafo generado hasta que los grafos resultantes sean completos, $\chi(G)$ es el tamaño del menor completo obtenido.

Algoritmo para coloreo de grafos

No se conoce un algoritmo polinomial para encontrar $\chi(G)$.
Veamos el siguiente algoritmo (conexión-contracción):

- Consideremos un grafo G con dos vértices no adyacentes a y b. La conexión G_1 se obtiene agregando la arista ab. La contracción G_2 se obtiene contrayendo $\{a, b\}$ en un solo vértice $c(a, b)$ que es vecino de $N(a) \cup N(b)$.
- Un coloreo de G en el cual a y b usan distintos colores es un coloreo de G_1, y viceversa. Un coloreo de G en el cual a y b usan el mismo color da lugar a un coloreo de G_2, y viceversa.
- Entonces, $\chi(G) = \text{mín}\{\chi(G_1), \chi(G_2)\}$.
- Si repetimos las operaciones en cada grafo generado hasta que los grafos resultantes sean completos, $\chi(G)$ es el tamaño del menor completo obtenido.

Ejemplo



Polinomio cromático

El polinomio cromático de un grafo G es una función $P_G(k)$ que para cada entero k da el número posible de k-coloreos de G.

- ¿Por qué esa función es un polinomio?
- **Ejemplo 1:** Si G es un árbol de n vértices, entonces:

$$P_G(k) = k(k-1)^{n-1}$$

- **Ejemplo 2:** Si G es K_n, entonces:

$$P_G(k) = k(k-1)(k-2)\dots(k-n+1)$$

Polinomio cromático

El polinomio cromático de un grafo G es una función $P_G(k)$ que para cada entero k da el número posible de k-coloreos de G.

- ¿Por qué esa función es un polinomio?
- **Ejemplo 1:** Si G es un árbol de n vértices, entonces:

$$P_G(k) = k(k-1)^{n-1}$$

- **Ejemplo 2:** Si G es K_n, entonces:

$$P_G(k) = k(k-1)(k-2)\ldots(k-n+1)$$

Polinomio cromático

El polinomio cromático de un grafo G es una función $P_G(k)$ que para cada entero k da el número posible de k-coloreos de G.

- ¿Por qué esa función es un polinomio?
- **Ejemplo 1:** Si G es un árbol de n vértices, entonces:

$$P_G(k) = k(k-1)^{n-1}$$

- **Ejemplo 2:** Si G es K_n, entonces:

$$P_G(k) = k(k-1)(k-2)\dots(k-n+1)$$

Polinomio cromático

Propiedad

$P_G(k) = P_{G_1}(k) + P_{G_2}(k)$, donde G_1 y G_2 son los grafos definidos en el algoritmo de conexión-contracción.

Polinomio cromático

Propiedad

$P_G(k) = P_{G_1}(k) + P_{G_2}(k)$, donde G_1 y G_2 son los grafos definidos en el algoritmo de conexión-contracción.

Corolario

$P_G(k)$ es un polinomio.

Polinomio cromático

Propiedad

$P_G(k) = P_{G_1}(k) + P_{G_2}(k)$, donde G_1 y G_2 son los grafos definidos en el algoritmo de conexión-contracción.

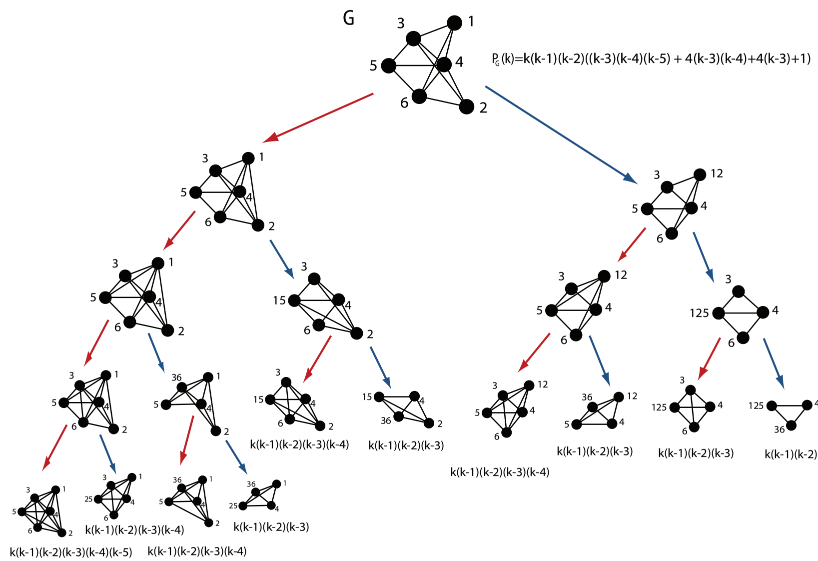
Corolario

$P_G(k)$ es un polinomio.

Ejercicio: Probar que el polinomio cromático de un ciclo C_n es

$$P_{C_n}(k) = (k-1)^n + (-1)^n(k-1)$$

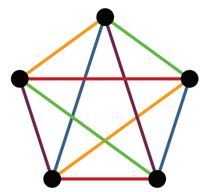
Ejemplo

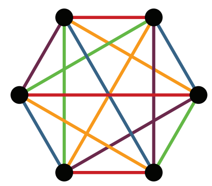


G
$P_G(k)=k(k-1)(k-2)((k-3)(k-4)(k-5)+4(k-3)(k-4)+4(k-3)+1)$
$k(k-1)(k-2)(k-3)(k-4)(k-5)$
$k(k-1)(k-2)(k-3)(k-4)$
$k(k-1)(k-2)(k-3)(k-4)$
$k(k-1)(k-2)(k-3)$
$k(k-1)(k-2)(k-3)(k-4)$
$k(k-1)(k-2)(k-3)$
$k(k-1)(k-2)(k-3)(k-4)$
$k(k-1)(k-2)(k-3)$
$k(k-1)(k-2)(k-3)$
$k(k-1)(k-2)$

Indice cromático

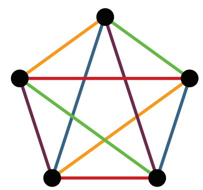
- El índice cromático $\chi'(G)$ de un grafo G es el menos número de colores necesarios para pintar las aristas de G de manera tal que dos aristas incidentes no tengan el mismo color.
- Claramente, $\chi'(G) \geq \Delta(G)$.

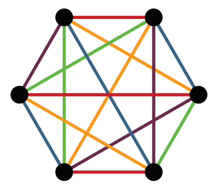




Indice cromático

- El índice cromático $\chi'(G)$ de un grafo G es el menos número de colores necesarios para pintar las aristas de G de manera tal que dos aristas incidentes no tengan el mismo color.
- Claramente, $\chi'(G) \geq \Delta(G)$.





Indice cromático

Propiedad

- $\chi'(K_n) = n - 1 = \Delta(K_n)$, si n es par
- $\chi'(K_n) = n = \Delta(K_n) + 1$, si n es impar

Indice cromático

Propiedad

- $\chi'(K_n) = n - 1 = \Delta(K_n)$, si n es par
- $\chi'(K_n) = n = \Delta(K_n) + 1$, si n es impar

Demo: A lo sumo $\lfloor \frac{n}{2} \rfloor$ aristas pueden usar el mismo color, por lo tanto $\chi'(G)\lfloor \frac{n}{2} \rfloor \geq m$. Como $m_{K_n} = \frac{n(n-1)}{2}$, para n impar $\chi'(K_n) \geq n$ y para n par $\chi'(K_n) \geq n - 1$.

Indice cromático

Propiedad

- $\chi'(K_n) = n - 1 = \Delta(K_n)$, si n es par
- $\chi'(K_n) = n = \Delta(K_n) + 1$, si n es impar

Demo: A lo sumo $\lfloor \frac{n}{2} \rfloor$ aristas pueden usar el mismo color, por lo tanto $\chi'(G)\lfloor \frac{n}{2} \rfloor \geq m$. Como $m_{K_n} = \frac{n(n-1)}{2}$, para n impar $\chi'(K_n) \geq n$ y para n par $\chi'(K_n) \geq n - 1$.

Vamos a construir coloreos con esa cantidad de colores respectivamente. Para n impar, dibujamos los vértices formando un polígono regular, pintamos un lado de cada color y cada diagonal del color de su lado paralelo.

Indice cromático

Propiedad

- $\chi'(K_n) = n - 1 = \Delta(K_n)$, si n es par
- $\chi'(K_n) = n = \Delta(K_n) + 1$, si n es impar

Demo: A lo sumo $\lfloor \frac{n}{2} \rfloor$ aristas pueden usar el mismo color, por lo tanto $\chi'(G)\lfloor \frac{n}{2} \rfloor \geq m$. Como $m_{K_n} = \frac{n(n-1)}{2}$, para n impar $\chi'(K_n) \geq n$ y para n par $\chi'(K_n) \geq n - 1$.

Vamos a construir coloreos con esa cantidad de colores respectivamente. Para n impar, dibujamos los vértices formando un polígono regular, pintamos un lado de cada color y cada diagonal del color de su lado paralelo. Para n par, sacamos un vértice v y pintamos el grafo K_{n-1}. En el coloreo resultante, cada vértice w es incidente a aristas de todos los colores salvo el color de su lado opuesto, y ese color se asigna a la arista vw (en el dibujo anterior, v es el vértice superior derecho). $\square$

Indice cromático

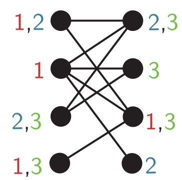
Teorema de Vizing (1964)

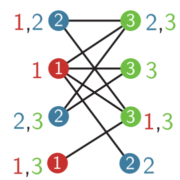
Sea G un grafo, entonces

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1.$$

Coloreo por listas

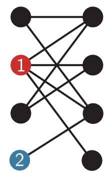
- En lugar de haber un conjunto de colores $1, \ldots, k$ disponibles para todos los vertices, cada vértice tiene su propia lista finita de colores admisibles.



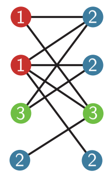


Precoloring Extension

- Consiste en extender un *k*-coloreo parcial a un *k*-coloreo total del grafo.



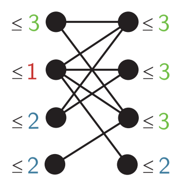
k=3

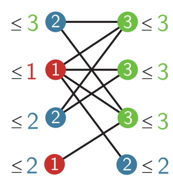


k=3

μ-coloreo

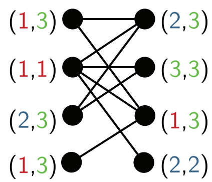
- En lugar de una cota k general, cada vértice tiene una cota propia y debe usar un color menor o igual a su cota.

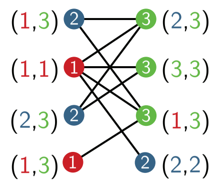




(γ, μ)-coloreo

- Ahora cada vértice tiene tanto una cota superior como inferior.





Jerarquía de problemas de coloreo

$$
\begin{tikzcd}[arrows=dash]
 & \text{coloreo por listas} \arrow[d, "<"] & \\
 & (\gamma,\mu)\text{-coloreo} \arrow[dl, "<"'] \arrow[dr, "\leq"] & \\
\text{PrExt} \arrow[dr, "<"'] & & \mu\text{-coloreo} \arrow[dl, "<"] \\
 & \text{k-coloreo} &
\end{tikzcd}
$$