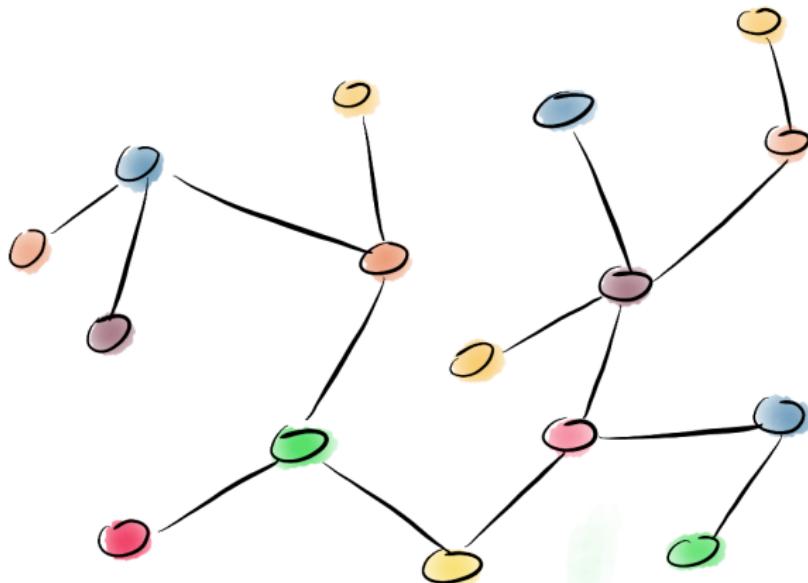


Árboles

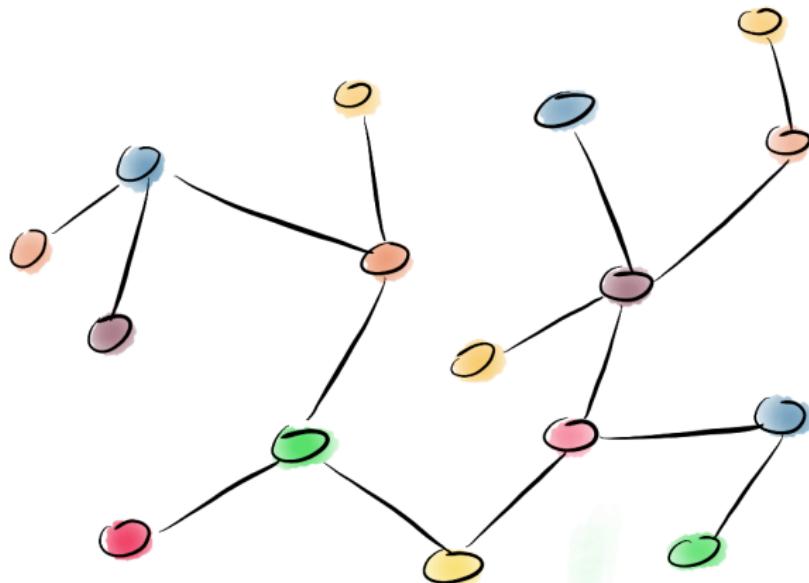
Algoritmos y Estructuras de Datos III

Departamento de Computación, FCEyN, Universidad de Buenos Aires



Árboles

Un **árbol** es un grafo conexo y acíclico (sin ciclos).



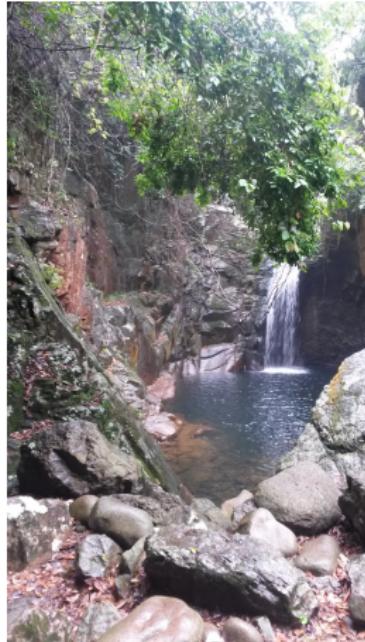
Árboles

Un **árbol** es un grafo conexo y acíclico (sin ciclos).

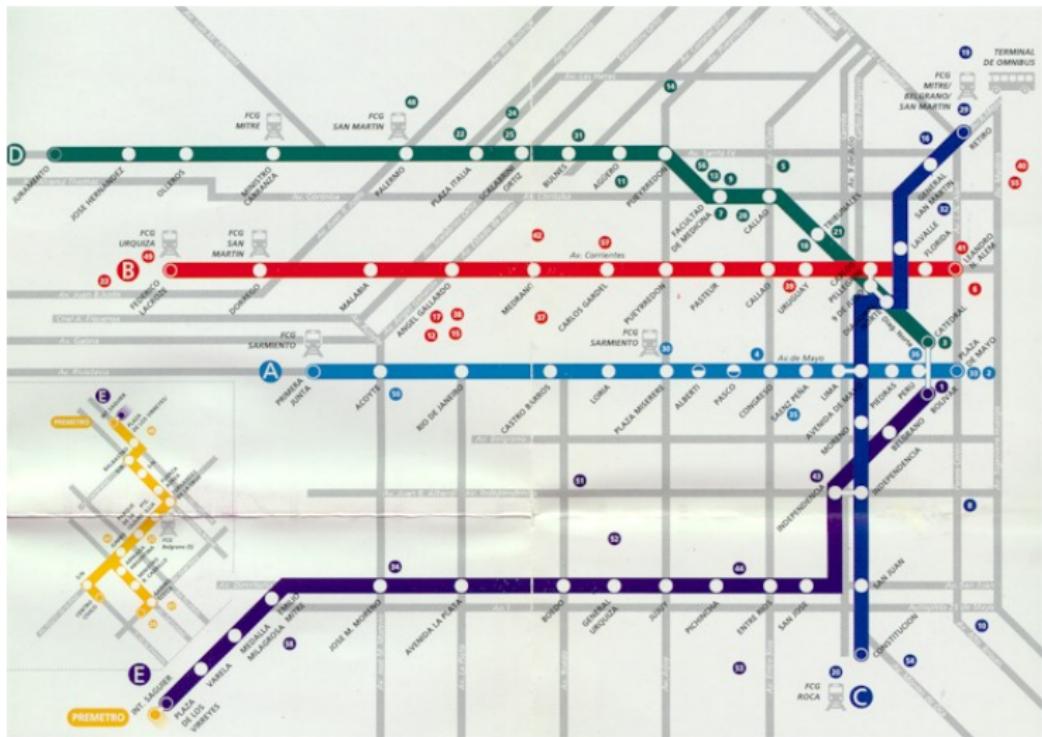


Árboles

Un **árbol** es un grafo conexo y acíclico (sin ciclos).



Árboles: aplicaciones

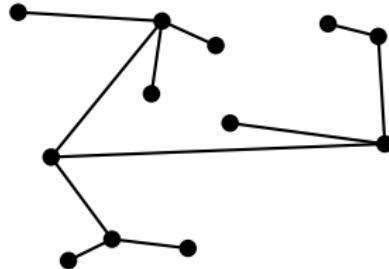


Árboles: aplicaciones



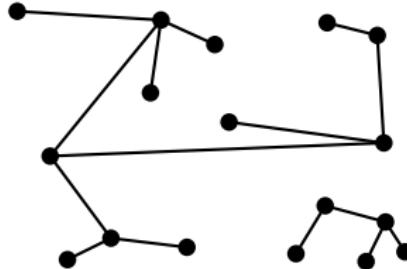
Árboles

- Un árbol es un grafo conexo y acíclico (sin ciclos).



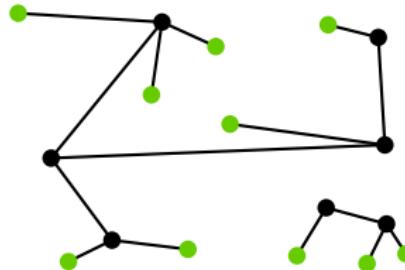
Árboles

- Un **árbol** es un grafo conexo y acíclico (sin ciclos).
- Un **bosque** es un grafo acíclico, o sea, una unión disjunta de árboles.



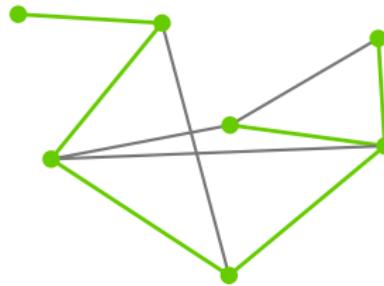
Árboles

- 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.



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.



Lema 1

Si $m_G > n_G - 1$ entonces G tiene un ciclo.

Lema 1

Si $m_G > n_G - 1$ entonces G tiene un ciclo.

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

Lema 1

Si $m_G > n_G - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n_G = 1$ o 2 , no puede pasar. Si $n_G = 3$, 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 $m_G > n_G - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n_G = 1$ o 2 , no puede pasar. Si $n_G = 3$, 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 $m_G > n_G - 1$ entonces G tiene un ciclo.

Demo: Por inducción. Si $n_G = 1$ o 2 , no puede pasar. Si $n_G = 3$, 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 . □

Lema 2

Si G es conexo, entonces $m_G \geq n_G - 1$.

Lema 2

Si G es conexo, entonces $m_G \geq n_G - 1$.

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

Lema 2

Si G es conexo, entonces $m_G \geq n_G - 1$.

Demo: Por inducción. Si $n_G = 1$ 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 $m_G \geq n_G - 1$.

Lema 2

Si G es conexo, entonces $m_G \geq n_G - 1$.

Demo: Por inducción. Si $n_G = 1$ 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 $m_G \geq n_G - 1$.

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 $m_G \geq n_G - 1$.

Demo: Por inducción. Si $n_G = 1$ 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 $m_G \geq n_G - 1$.

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

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



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 $m_G = n_G - 1$.
4. G es acíclico y $m_G = n_G - 1$.

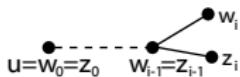
G es un árbol \Leftrightarrow todo par de vértices de G está unido por un único camino

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, $P_1 : u = w_0, w_1, \dots, w_k = v$ y $P_2 : u = z_0, z_1, \dots, z_r = v$. 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, $P_1 : u = w_0, w_1, \dots, w_k = v$ y $P_2 : u = z_0, z_1, \dots, z_r = v$. Sea i el primer índice tal que $w_i \neq z_i$. Entonces $i > 0$, y $w_{i-1} = z_{i-1}$.



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

G es un árbol $\Leftrightarrow G$ es conexo y $m_G = n_G - 1$

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

G es un árbol $\Leftrightarrow G$ es conexo y $m_G = n_G - 1$

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

\Leftarrow) G es conexo, probemos por inducción que es un árbol. Si $n_G = 1$, vale.
Supongamos $n_G > 1$. Por propiedad de la suma de grados, G tiene al menos un vértice v de grado menor o igual que uno.

G es un árbol $\Leftrightarrow G$ es conexo y $m_G = n_G - 1$

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

\Leftarrow) G es conexo, probemos por inducción que es un árbol. Si $n_G = 1$, vale. Supongamos $n_G > 1$. 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.

G es un árbol $\Leftrightarrow G$ es conexo y $m_G = n_G - 1$

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

\Leftarrow) G es conexo, probemos por inducción que es un árbol. Si $n_G = 1$, vale. Supongamos $n_G > 1$. 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

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

G es un árbol $\Leftrightarrow G$ es conexo y $m_G = n_G - 1$

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

\Leftarrow) G es conexo, probemos por inducción que es un árbol. Si $n_G = 1$, vale. Supongamos $n_G > 1$. 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). □

G es un árbol $\Leftrightarrow G$ es acíclico y $m_G = n_G - 1$

Demo: \Rightarrow) Si G es un árbol entonces es acíclico y por Lema 1,
 $m_G \leq n_G - 1$.

G es un árbol $\Leftrightarrow G$ es acíclico y $m_G = n_G - 1$

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

G es un árbol $\Leftrightarrow G$ es acíclico y $m_G = n_G - 1$

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

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

G es un árbol $\Leftrightarrow G$ es acíclico y $m_G = n_G - 1$

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

$\Leftarrow)$ G es acíclico. Supongamos que tiene t componentes conexas G_1, \dots, 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. □

¿Cuántas aristas tiene un bosque de c componentes conexas?

¿Cuántas aristas tiene un bosque de c componentes conexas?

¿Cuál es el mínimo de aristas que tiene un grafo con c componentes conexas?

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 $v_0 w_0$.

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_0w_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.

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_0w_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.

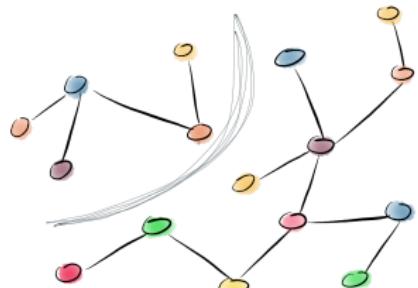
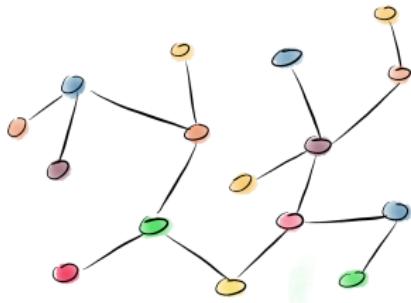
Teorema

Todo árbol no trivial tiene al menos dos hojas.

Demo: Si T es un árbol no trivial, por ser conexo tiene una arista v_0w_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 . □

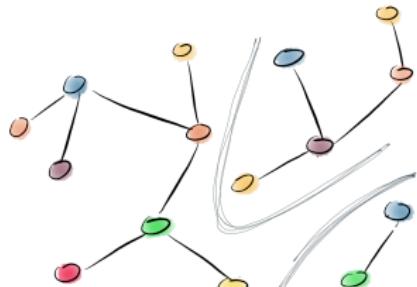
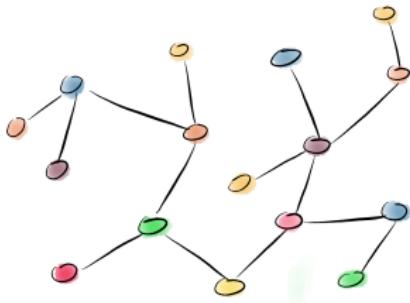
Teorema

1. Toda arista de un árbol es un puente.



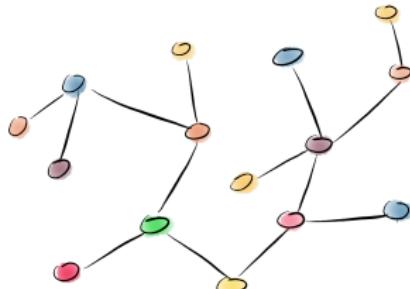
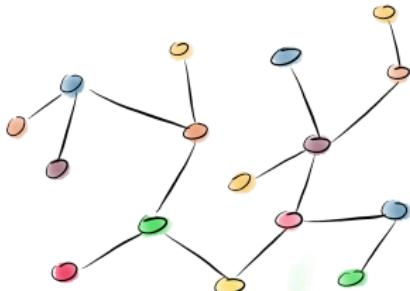
Teorema

1. Toda arista de un árbol es un puente.
2. Un vértice de un árbol no trivial es un punto de corte si y sólo si 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 si y sólo si 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 si y sólo si 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 si y sólo si 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 une 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 si y sólo si 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 une 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 .

Teorema

1. Toda arista de un árbol es un puente.
2. Un vértice de un árbol no trivial es un punto de corte si y sólo si 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 une v y w en $T - e$.

2. En cualquier grafo, una hoja nunca puede ser punto de corte.

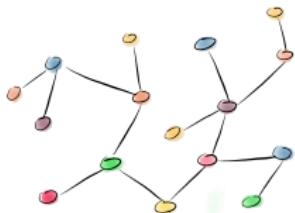
Sea T un árbol no trivial y v vértice de T que no es hoja.

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 $T - v$. □

Teorema

Son equivalentes:

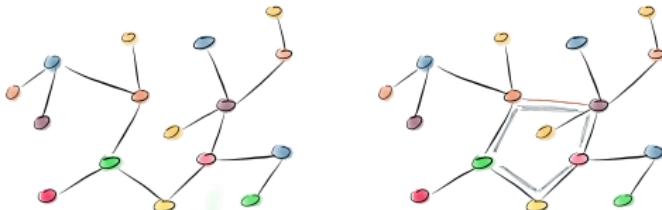
1. G es un árbol.



Teorema

Son equivalentes:

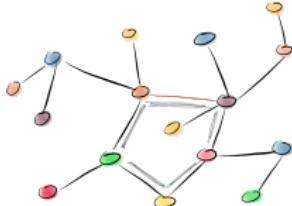
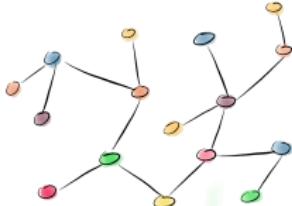
1. G es un árbol.
2. G es un grafo sin ciclos, pero si se agrega una arista e a G resulta un grafo con exactamente un ciclo, y ese ciclo contiene a e (es acíclico *maximal* respecto de aristas).



Teorema

Son equivalentes:

1. G es un árbol.
2. G es un grafo sin ciclos, pero si se agrega una arista e a G resulta un grafo con exactamente un ciclo, y ese ciclo contiene a e (es acíclico *maximal* respecto de aristas).
3. G es conexo, pero si se quita cualquier arista a G queda un grafo no conexo (es conexo *minimal* respecto de aristas).



Teorema

Son equivalentes:

1. G es un árbol.
2. G es un grafo sin ciclos, pero si se agrega una arista e a G resulta un grafo con exactamente un ciclo, y ese ciclo contiene a e (es acíclico *maximal* respecto de aristas).
3. G es conexo, pero si se quita cualquier arista a G queda un grafo no conexo (es conexo *minimal* respecto de aristas).

Lema

Sea $G = (V, E)$ un grafo conexo y $e \in E$. $G - e$ es conexo si y sólo si e pertenece a un ciclo de G .

Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Teorema

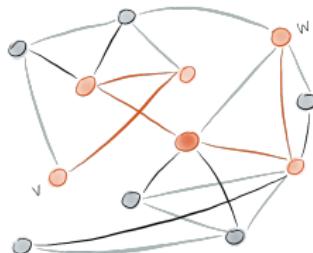
Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) Sea T un a.g. de G . Sean v, w vértices de G .

Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) 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.

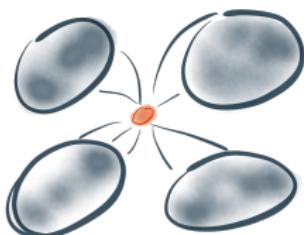
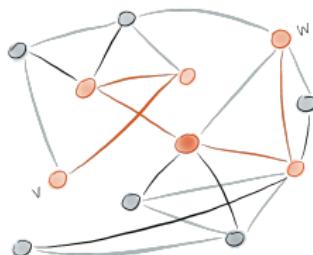


Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) 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.

\Rightarrow) Por inducción. Si G es trivial, G es un a.g. de si mismo. Si no, sea v un vértice de G .

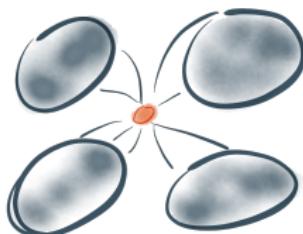
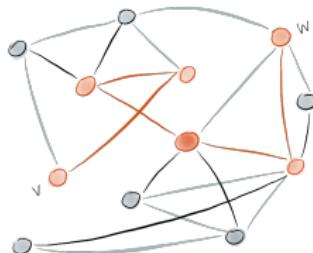


Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) 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.

\Rightarrow) 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 .

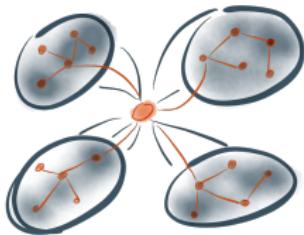
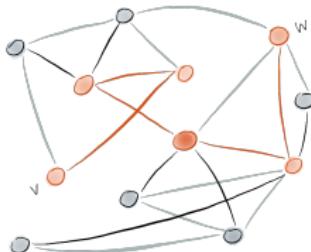


Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) 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.

\Rightarrow) 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 $v_i v$.

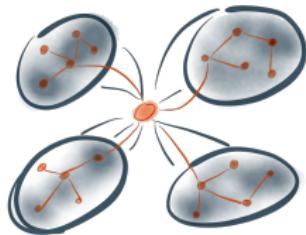
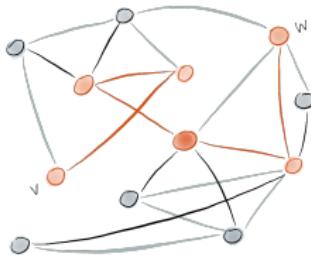


Teorema

Un grafo es conexo si y sólo si admite un árbol generador.

Idea de demo: \Leftrightarrow) 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.

\Rightarrow) 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_iv . 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 .

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.

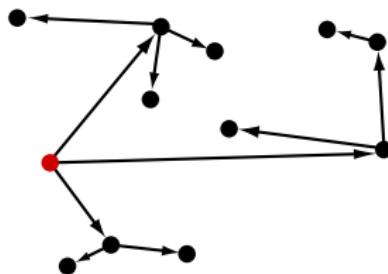
Corolario

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. □

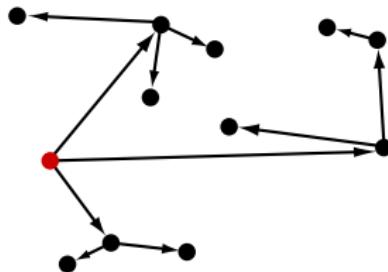
Árboles dirigidos

- 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.



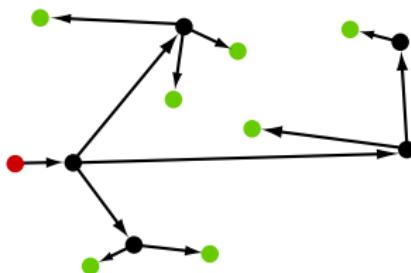
Árboles dirigidos

- 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 .



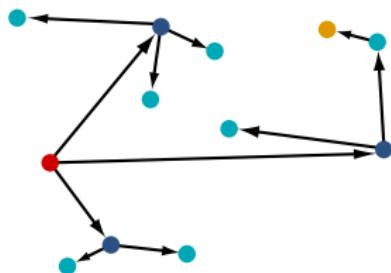
Árboles dirigidos

- 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 parente, salvo la raíz que no tiene. Se llaman **hojas** los vértices sin hijos.



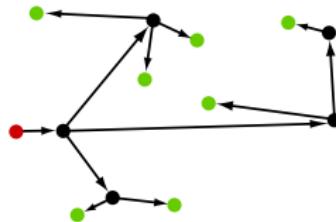
Árboles dirigidos

- 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 parente, 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.



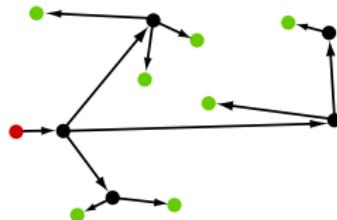
Árboles dirigidos

- Los vértices **internos** de un árbol son aquellos que no son ni hojas ni la raíz.



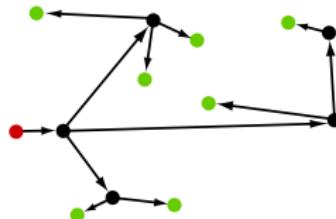
Árboles dirigidos

- Los vértices **internos** de un árbol son aquellos que no son ni hojas ni la raíz.
- Un árbol se dice (exactamente) ***m*-ario** si todos sus vértices internos tienen grado (exactamente) a lo sumo $m + 1$ y la raíz (exactamente) a lo sumo m .



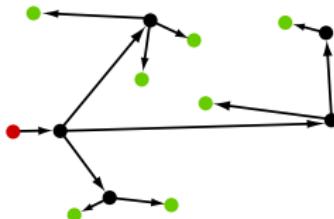
Árboles dirigidos

- Los vértices **internos** de un árbol son aquellos que no son ni hojas ni la raíz.
- Un árbol se dice (exactamente) ***m*-ario** si todos sus vértices internos tienen grado (exactamente) a lo sumo $m + 1$ y la raíz (exactamente) a lo sumo m .
- La **altura *h*** de un árbol es la longitud desde la raíz al vértice más lejano.



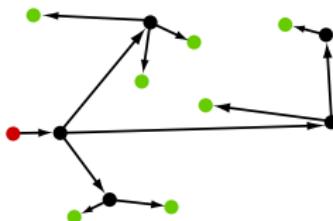
Árboles dirigidos

- Los vértices **internos** de un árbol son aquellos que no son ni hojas ni la raíz.
- Un árbol se dice (exactamente) ***m*-ario** si todos sus vértices internos tienen grado (exactamente) a lo sumo $m + 1$ y la raíz (exactamente) a lo sumo m .
- La **altura *h*** de un árbol es la longitud desde la raíz al vértice más lejano.
- Un árbol se dice **balanceado** si todas sus hojas están a nivel h o $h - 1$.



Árboles dirigidos

- Los vértices **internos** de un árbol son aquellos que no son ni hojas ni la raíz.
- Un árbol se dice (exactamente) ***m*-ario** si todos sus vértices internos tienen grado (exactamente) a lo sumo $m + 1$ y la raíz (exactamente) a lo sumo m .
- La **altura *h*** de un árbol es la longitud desde la raíz al vértice más lejano.
- Un árbol se dice **balanceado** si todas sus hojas están a nivel h o $h - 1$.
- Un árbol se dice **balanceado completo** si todas sus hojas están a nivel h .



Árboles dirigidos

¿Cuántos vértices tiene en total un árbol exactamente m -ario que tiene i vértices internos?

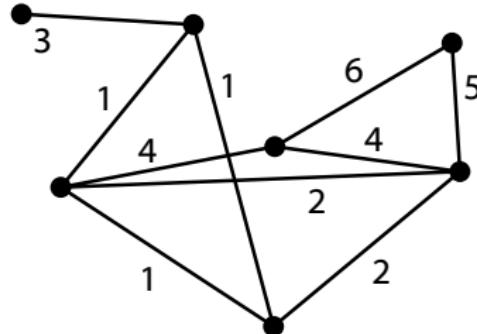
Árboles dirigidos

Teorema

1. Un árbol m -ario de altura h tiene a lo sumo m^h hojas.
2. Un árbol m -ario con l hojas tiene $h \geq \lceil \log_m l \rceil$.
3. Si T es un árbol exactamente m -ario balanceado entonces $h = \lceil \log_m l \rceil$.

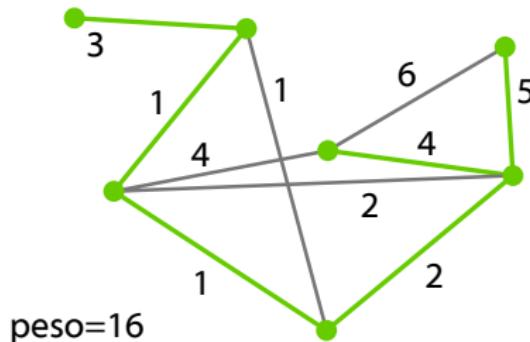
Árbol generador mínimo

- Un **grafo pesado** es un grafo tal que sus aristas tienen asociado un peso.



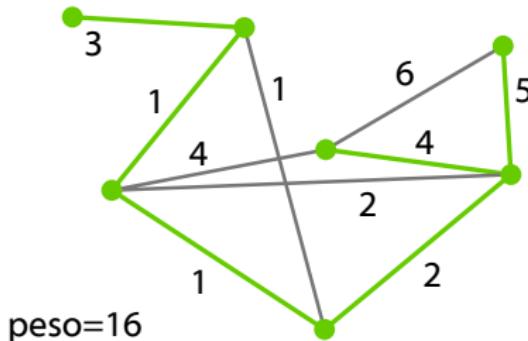
Árbol generador mínimo

- 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.



Árbol generador mínimo

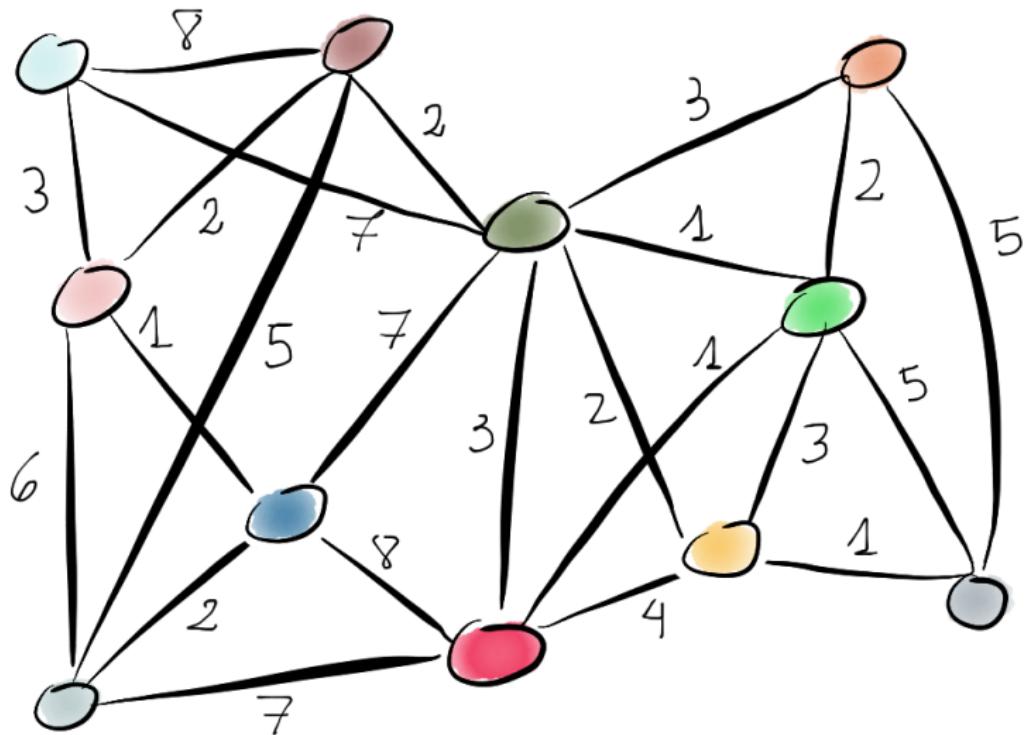
- Un **grafo pesado** es un grafo tal que sus aristas tienen asociado un peso.
- El **peso** de un subgrafo es la suma de los pesos de sus aristas.
- Un **árbol generador mínimo** en un grafo pesado es un árbol generador de peso mínimo.



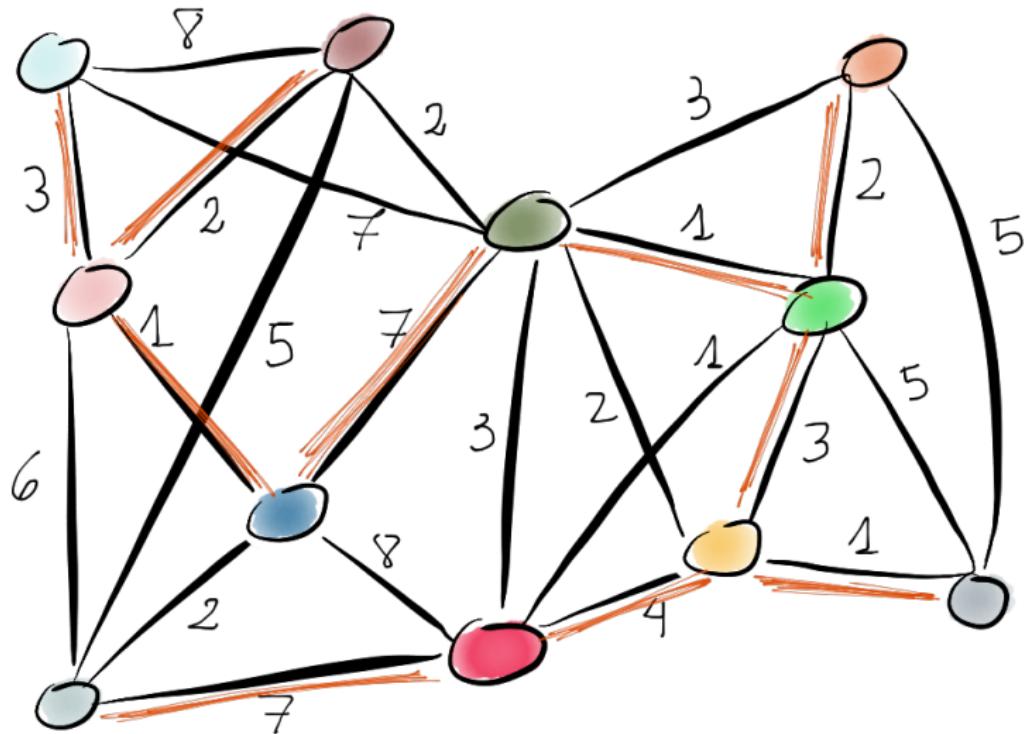
Ejemplo de aplicación

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?

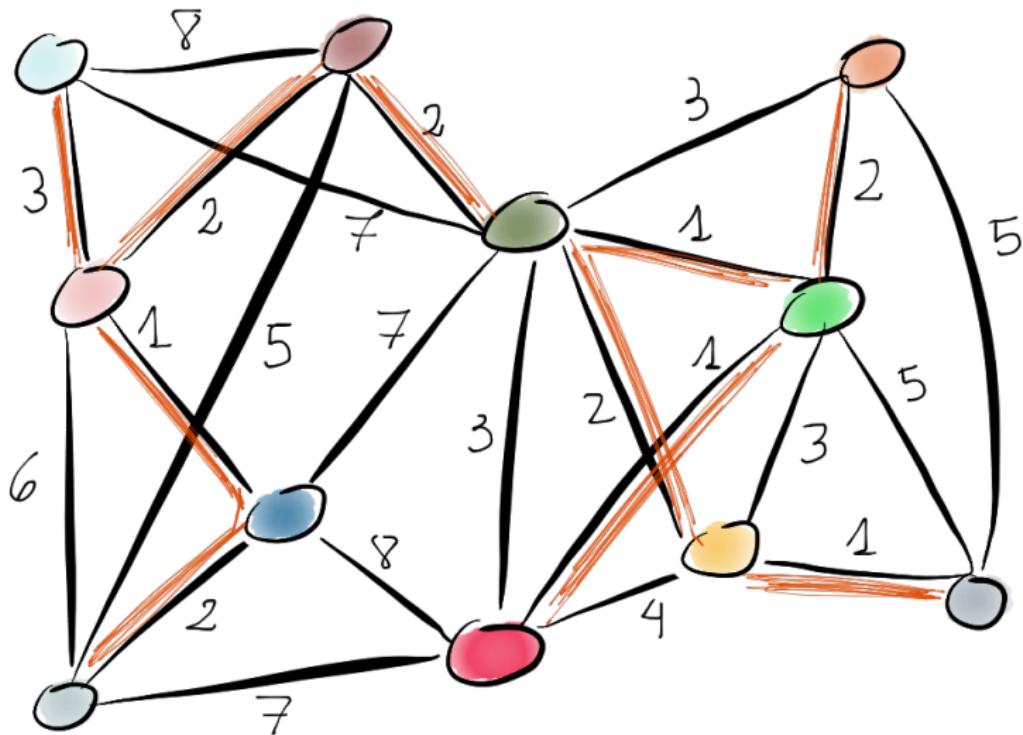
Calculando un AGM “a ojo”



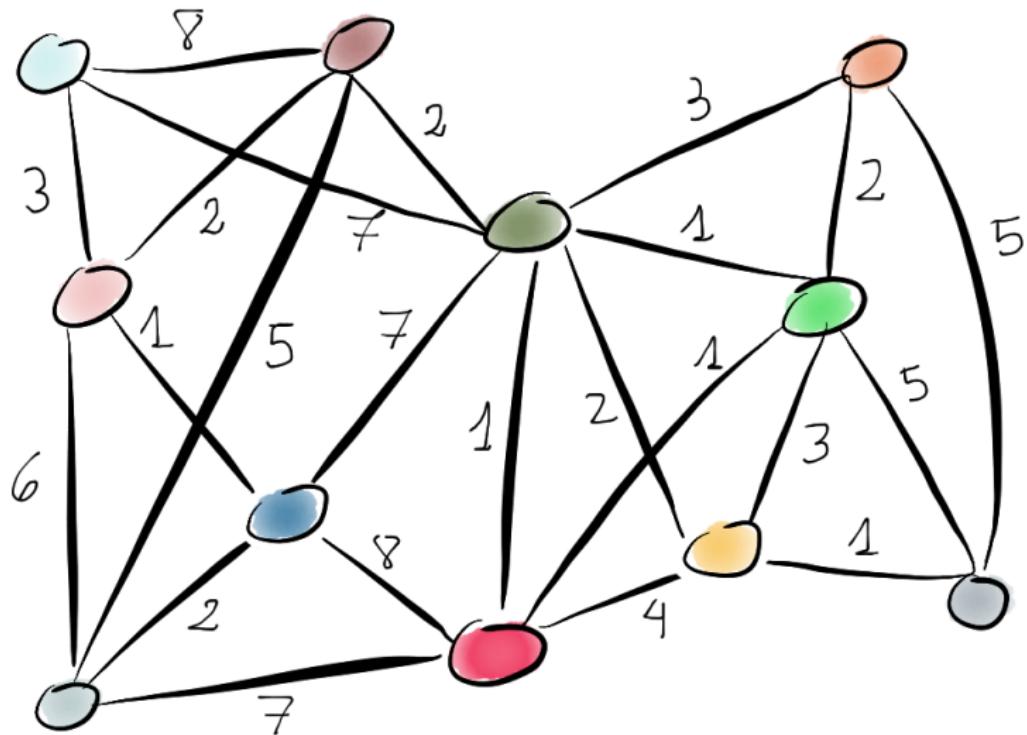
¿Es un árbol generador mínimo?



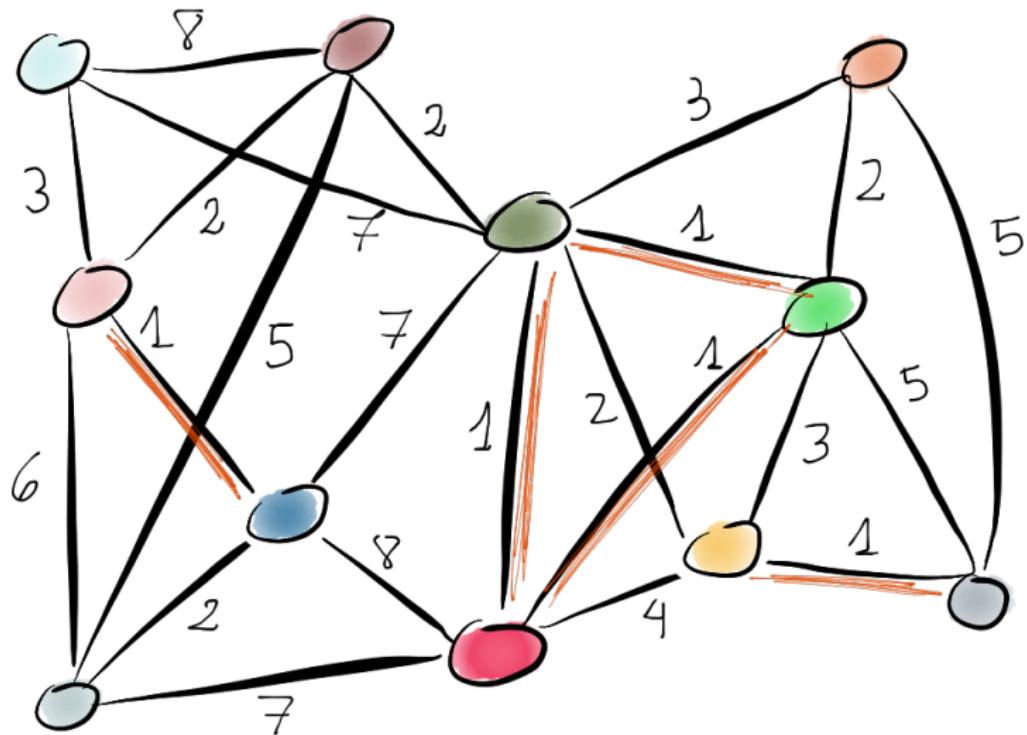
¿Y éste?



Calculando un AGM “a ojo”



Calculando un AGM “a ojo”



Calculando un AGM “a ojo”

De las aristas de peso mínimo, ¿cuántas puedo usar a lo sumo?

Calculando un AGM “a ojo”

De las aristas de peso mínimo, ¿cuántas puedo usar a lo sumo? ¿Y de las de peso siguiente?

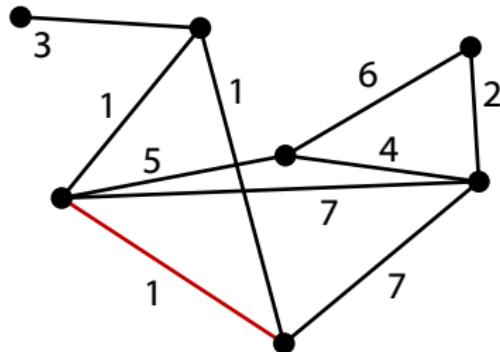
Algoritmo de Kruskal para AGM

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.

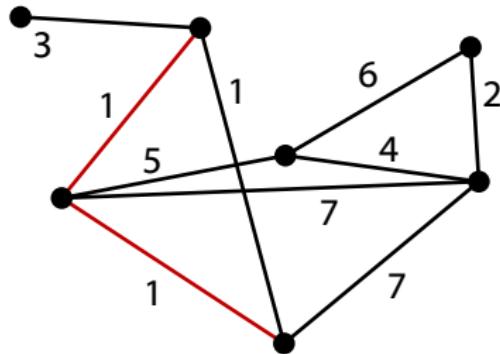
Ej:



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.

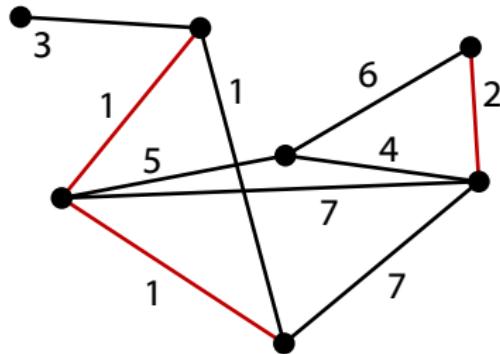
Ej:



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.

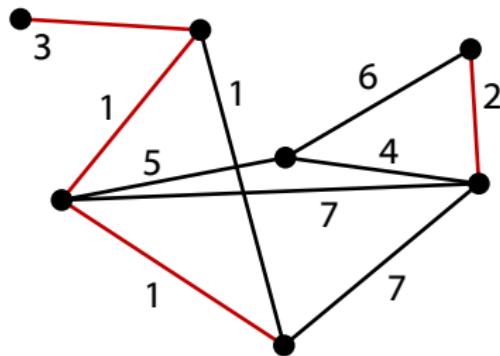
Ej:



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.

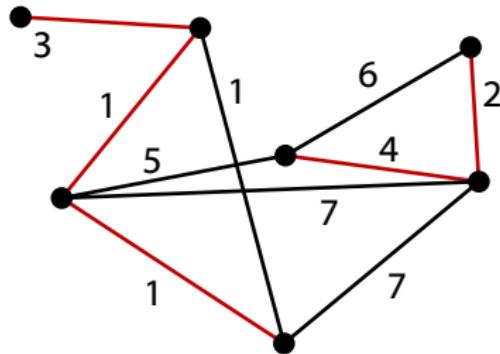
Ej:



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.

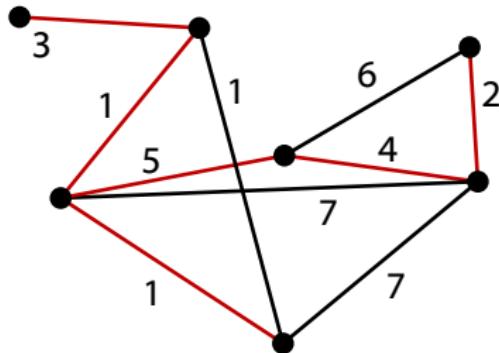
Ej:



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.

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

Demostración de que Kruskal construye un AGM

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

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

Demostración de que Kruskal construye un AGM

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

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

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

Demostración de que Kruskal construye un AGM

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

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

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

Luego T_K es un árbol generador mínimo. □

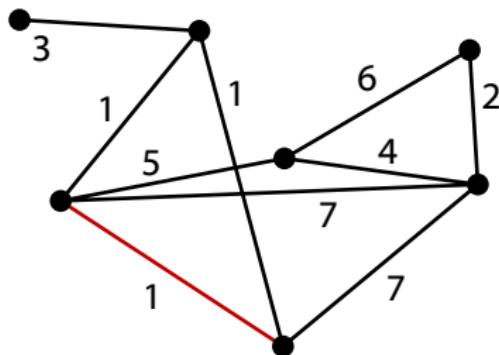
Algoritmo de Prim para AGM

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 $W = V(G)$.

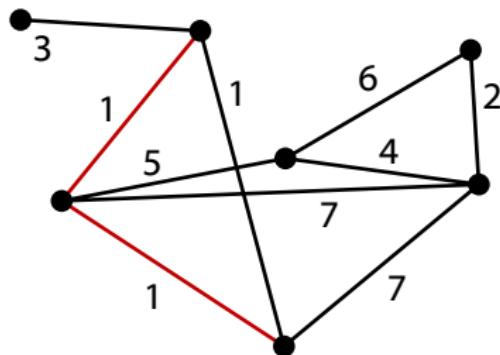
Ej:



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 $W = V(G)$.

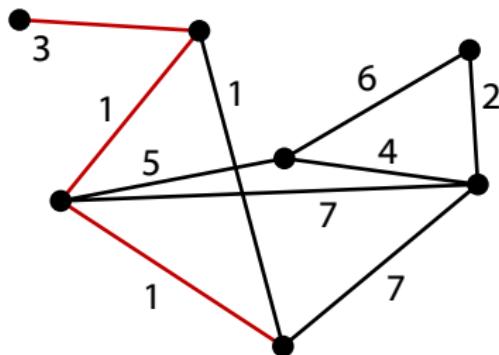
Ej:



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 $W = V(G)$.

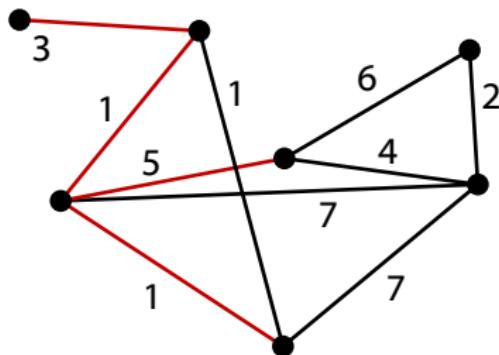
Ej:



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 $W = V(G)$.

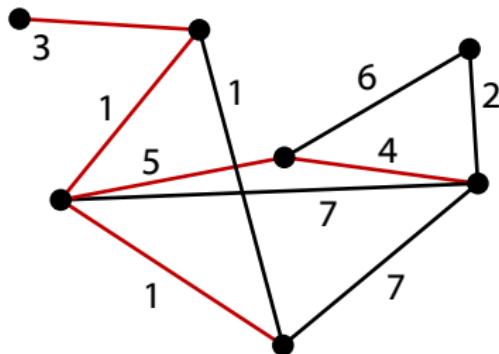
Ej:



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 $W = V(G)$.

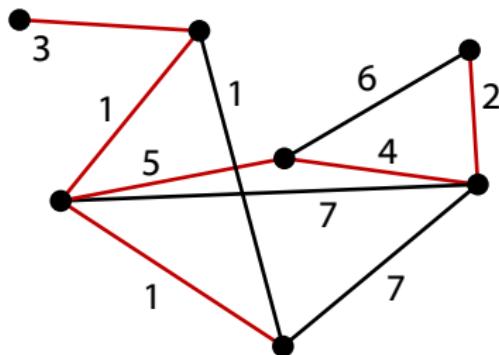
Ej:



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.

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 $\ell(T)$ como el máximo $k \leq n$ tal que $\forall j < k, e_j \in T$.

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 $\ell(T)$ como el máximo $k \leq n$ tal que $\forall j < k, e_j \in T$.

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

Demostración de que Prim construye un AGM

- Si $\ell(T) = 1$, como e_1 es de peso mínimo, $\text{peso}(e_1) \leq \text{peso}(e)$
 $\forall e \in C$. Luego $T' = T - e \cup \{e_{\ell(T)}\}$ es un árbol generador de G de peso menor o igual a T y $\ell(T') > \ell(T)$, absurdo.

Demostración de que Prim construye un AGM

- Si $\ell(T) = 1$, como e_1 es de peso mínimo, $\text{peso}(e_1) \leq \text{peso}(e)$ $\forall e \in C$. Luego $T' = T - e \cup \{e_{\ell(T)}\}$ es un árbol generador de G de peso menor o igual a T y $\ell(T') > \ell(T)$, absurdo.
- Si $\ell(T) > 1$, sea V_1 el conjunto de extremos de las aristas $e_1, \dots, e_{\ell(T)-1}$ y $V_2 = V - V_1$. Por la forma de elegir las aristas en Prim, $e_{\ell(T)}$ 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 partitionar entre los de V_1 y los de V_2 , ambos conjuntos son no vacíos y C es conexo). Entonces $\text{peso}(e_{\ell(T)}) \leq \text{peso}(e)$ y $T - e \cup \{e_{\ell(T)}\}$ es un árbol generador de peso menor o igual a T y $\ell(T') > \ell(T)$ (e no es ninguna de las e_i con $i < \ell(T)$ porque esas tienen ambos extremos en V_1 , por definición de V_1), absurdo.

Demostración de que Prim construye un AGM

- Si $\ell(T) = 1$, como e_1 es de peso mínimo, $\text{peso}(e_1) \leq \text{peso}(e)$ $\forall e \in C$. Luego $T' = T - e \cup \{e_{\ell(T)}\}$ es un árbol generador de G de peso menor o igual a T y $\ell(T') > \ell(T)$, absurdo.
- Si $\ell(T) > 1$, sea V_1 el conjunto de extremos de las aristas $e_1, \dots, e_{\ell(T)-1}$ y $V_2 = V - V_1$. Por la forma de elegir las aristas en Prim, $e_{\ell(T)}$ 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 partitionar entre los de V_1 y los de V_2 , ambos conjuntos son no vacíos y C es conexo). Entonces $\text{peso}(e_{\ell(T)}) \leq \text{peso}(e)$ y $T - e \cup \{e_{\ell(T)}\}$ es un árbol generador de peso menor o igual a T y $\ell(P)$ mayor a $\ell(T)$ (e no es ninguna de las e_i con $i < \ell(T)$ porque esas tienen ambos extremos en V_1 , por definición de V_1), absurdo.

Luego P es un árbol generador mínimo. □

Enumeración

- ¿Cuántos grafos diferentes hay que tengan ciertas características?

Enumeración

- ¿Cuántos grafos diferentes hay que tengan ciertas características?
- ¿Cuántos grafos conexos con 8 vértices y 2 circuitos?

Enumeración

- ¿Cuántos grafos diferentes hay que tengan ciertas características?
- ¿Cuántos grafos conexos con 8 vértices y 2 circuitos?
- ¿Cuántos subgrafos de un cierto tipo de un grafo dado hay?

Enumeración

- ¿Cuántos grafos diferentes hay que tengan ciertas características?

¿Cuántos grafos conexos con 8 vértices y 2 circuitos?

- ¿Cuántos subgrafos de un cierto tipo de un grafo dado hay?

¿Cuántos árboles generadores tiene un grafo?

¿Cuántos circuitos simples?

¿Cuántos subgrafos completos maximales?

Enumeración

- ¿Cuántos grafos diferentes hay que tengan ciertas características?
 - ¿Cuántos grafos conexos con 8 vértices y 2 circuitos?
 - ¿Cuántos subgrafos de un cierto tipo de un grafo dado hay?
 - ¿Cuántos árboles generadores tiene un grafo?
 - ¿Cuántos circuitos simples?
 - ¿Cuántos subgrafos completos maximales?

En general son problemas difíciles.

Ejemplos de algunas cosas que se *pueden* calcular

¿Cuántos grafos rotulados distintos de n vértices y m aristas hay?

¿Cuántos grafos rotulados distintos de n vértices hay?

Teorema de Cayley (1875)

El número de árboles rotulados distintos de n vértices es n^{n-2} , para $n \geq 2$.

Demo: Por inducción en n usando los algoritmos inversos de codificación y decodificación de *Prüfer* para establecer una biyección entre el conjunto de árboles rotulados de n vértices y el conjunto de palabras de $n - 2$ símbolos usando el alfabeto $\{1, \dots, n\}$. □

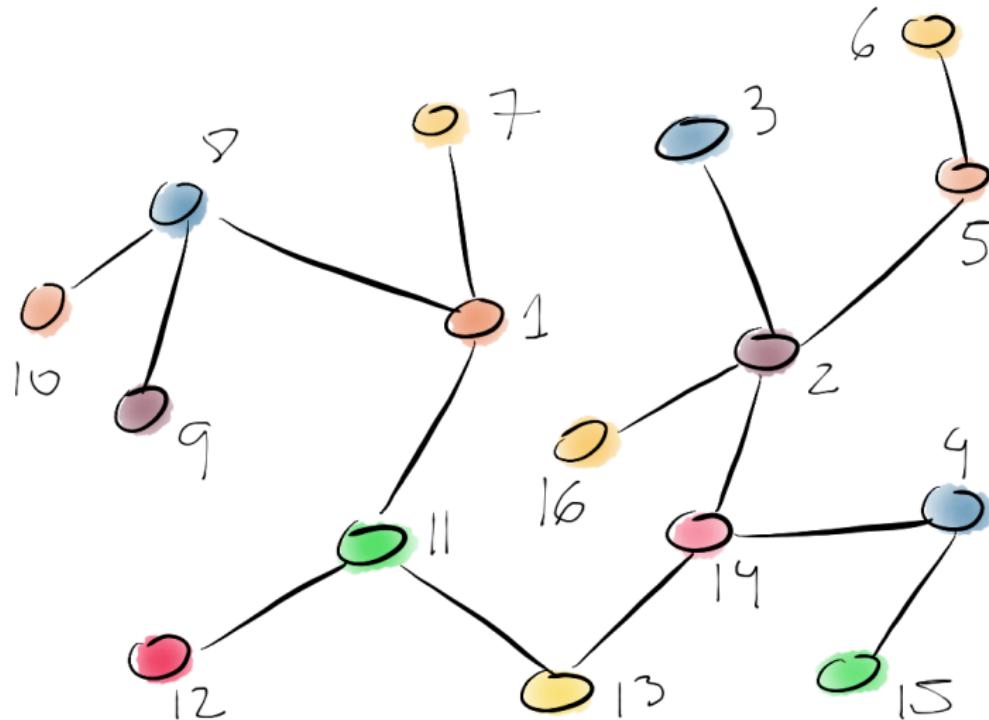
Algoritmo de codificación de Prüfer

Dado un árbol de n vértices rotulado con los números de 1 a n , construye una palabra de longitud $n - 2$ usando el alfabeto $\{1, \dots, n\}$.

Entrada: Un árbol T rotulado de n vértices, $n \geq 3$.

```
para  $i = 1, \dots, n - 2$  hacer
    elegir  $v$  la hoja de menor rótulo de  $T$ 
    poner como  $s_i$  el rótulo del vértice adyacente a  $v$ 
     $T := T \setminus \{v\}$ 
retornar palabra  $s_1 \dots s_{n-2}$ 
```

Algoritmo de codificación de Prüfer



Algoritmo de codificación de Prüfer

Proposición

Si d_k es el número de ocurrencias del número k en la palabra retornada por el algoritmo de codificación de Prüfer, entonces el grado del vértice con rótulo k en el árbol T es $d_k + 1$.

Algoritmo de decodificación de Prüfer

Dada una palabra de longitud $n - 2$ usando el alfabeto $\{1, \dots, n\}$ construye un árbol de n vértices rotulado con los números de 1 a n .

Entrada: Una palabra $P = s_1 \dots s_{n-2}$.

$$L := \{1, \dots, n\}$$

$$V_T := \{1, \dots, n\}$$

$$E_T := \emptyset$$

$$P := s_1 \dots s_{n-2}$$

para $i = 1, \dots, n - 2$ **hacer**

Sea k el menor número de L que no está en P

$$E_T := E_T \cup \{(k, s_i)\}$$

$$L := L \setminus \{k\}$$

$$P := s_{i+1} \dots s_{n-2}$$

$$E_T := E_T \cup \{(u, v)\}, \text{ con } u, v \in L$$

retornar $T = (V_T, E_T)$

Algoritmo de decodificación de Prüfer

Proposición

El algoritmo de decodificación de Prüfer define una función del conjunto de palabras de longitud $n - 2$ usando el alfabeto $\{1, \dots, n\}$ sobre el conjunto de árboles rotulados de n vértices.

Ejercicio para entregar

Ejercicio

Demostrar que el algoritmo de decodificación es el inverso del de codificación.