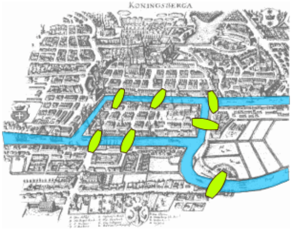
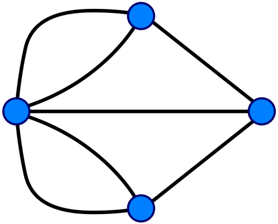
Ampliación de Matemáticas
Teoría de Grafos

Grafos:
Isomorfía, conexión y grados





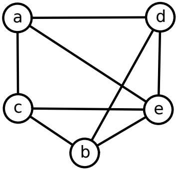
La Teoría de Grafos comenzó en 1735 al resolver Leonard Euler el problema de los puentes de Königsberg.

Los grafos permiten modelizar situaciones en lo que lo único que importa es si los objetos están relacionados.

Tienen multitud de aplicaciones:

- Redes
- Diagramas de flujo
- Bases de Datos
- etc

Definición de grafo



Un **grafo** (V, E) está formado por

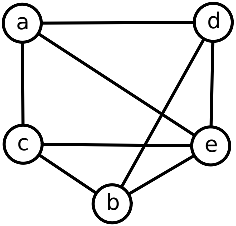
- un **conjunto de vértices** V
- un **conjunto de aristas** E

Las aristas se determinan por los vértices que conectan.

Decimos que dos vértices $u, v \in V$ son **adyacentes** si existe una arista $e = \{u, v\}$ que los conecta.

Así, decimos que e es **incidente** con u y v y que u, v son los **extremos** de e.

Definición de grafo

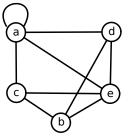


En el grafo de la izquierda:

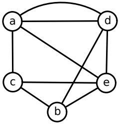
$$V = \{a, b, c, d, e\}$$

$$\begin{aligned} E = \{ & \{a, c\}, \{a, d\}, \{a, e\}, \\ & \{b, c\}, \{b, d\}, \{b, e\}, \\ & \{c, e\}, \{d, e\}\} \end{aligned}$$

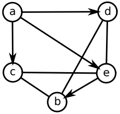
Generalizaciones del concepto de grafo



Si admitimos vértices conectados con ellos mismos: **pseudografo**. En dicho caso se denominan **lazos** a las aristas que conectan un vértice consigo mismo.

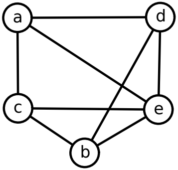


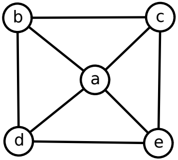
Si hay más de una conexión entre dos vértices: **multigrafo**. Se denominan **paralelas** a las aristas que conectan los mismos vértices.



Si los pares de vértices son ordenados: **digrafo** o **grafo dirigido**

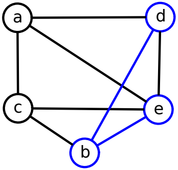
Grafos isomorfos

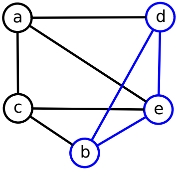




Dos grafos $G = (V, E)$ y $G = (V', E')$ son **isomorfos** si renombrando los vértices y aristas del primero, obtenemos el segundo

Subgrafos

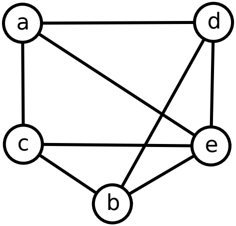




Denominamos **subgrafo** de un grafo a un grafo obtenido tomando un subconjunto de vértices y un subconjunto de aristas y todos los extremos de estas.

Denominamos **subgrafo completo** de un grafo a un grafo obtenido tomando un subconjunto de vértices y todas las aristas cuyos extremos sean esos vértices.

Representación con listas



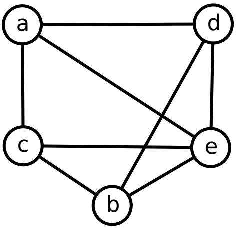
Lista de incidencia: Se enumeran las aristas

$$\{\{a,c\},\{a,d\},\{a,e\},\{b,c\},\{b,d\},\{b,e\},\{c,e\},\{d,e\}\}$$

Lista de adyacencia: Se enumeran los vértices adyacentes:

$$\{\{a \to c,d,e\},\{b \to c,d,e\},\{c \to a,b,e\},\{d \to a,b,e\}\ \{e \to a,b,c,d\}\}$$

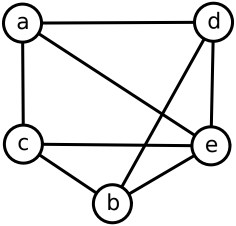
Representación con matrices



Matriz de incidencia: Se representan las aristas. -1 del vértice que sale y 1 al vértice que llega.

$$\begin{array}{ccccc} a & b & c & d & e \end{array}$$
$$\begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Representación con matrices



Matriz de adyacencia: En la posición (i, j) hay un 1 si las aristas i, j son adyacentes.

$$\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}
\end{array}$$

Isomorfía a partir de la matriz de adyacencia

Sean $M_G, M_{G'}$ matrices de adyacencia de los grafos G y G'.

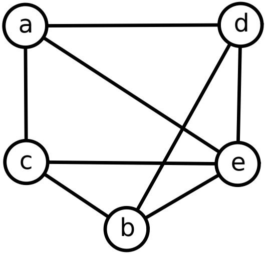
Teorema

Los grafos G y G' son isomorfos si y solo si existe una matriz de permutaciones P tal que $PM_G = M_{G'}P$, o, equivalentemente, $P^t M_G P = M_{G'}$.

$$M_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array} \qquad M_{G'} = \begin{array}{c} \\ a' \\ b' \\ c' \\ d' \\ e' \end{array} \begin{array}{c} \begin{array}{ccccc} a' & b' & c' & d' & e' \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

P es la permutación $\{1,2,3,4,5\} \rightsquigarrow \{5,2,3,4,1\}$

Grado de un vértice



Denominamos **grado** de un vértice al número de aristas que lo tienen como extremo.

En el grafo de la figura:

$$\mathrm{gr}(a) = 3, \ \mathrm{gr}(e) = 4$$

Se denomina **secuencia de grados** de un grafo a la lista (decreciente) de los grados del grafo.

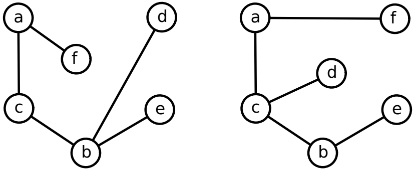
$$(4, 3, 3, 3, 3)$$

Secuencia de grados

Proposición

Si dos grafos son isomorfos, entonces tienen la misma secuencia de grados.

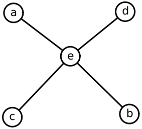
Dos grafos pueden tener la misma secuencia de grados y no ser isomorfos.

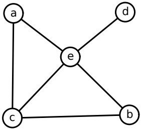


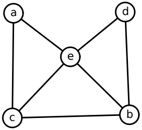
Propiedades de la secuencia de grados

- La suma de los grados es igual al doble del número de aristas.
- En particular, la suma de los grados es un número par.
- **Algoritmo de Havel-Hakimi:** Para determinar si una secuencia de números naturales $n_1 \geq n_2 \geq \ldots \geq n_k \geq 0$ es la secuencia de grados de un grafo, debemos usar el siguiente algoritmo:
  1. Si todos los elementos son cero, terminamos con respuesta afirmativa. En caso contrario,
  2. Eliminamos n_1 de la secuencia y restamos 1 a (los primeros) n_1 elementos.
  3. Si algún entero toma valor negativo, terminamos con respuesta negativa.
  4. Reordenamos la secuencia de mayor a menor.
  5. Volvemos al primer paso.

Algoritmo de Havel-Hakimi







Tomemos $(4,3,3,2,2)$. Asignamos a cada grado un nombre de vértice:

$$(\overset{e}{4},\overset{c}{3},\overset{b}{3},\overset{d}{2},\overset{a}{2})$$

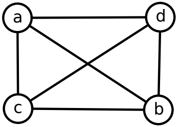
Unimos el vértice eliminado con los vértices a los que restamos.

$$(\overset{c}{2},\overset{b}{2},\overset{a}{1},\overset{d}{1})$$

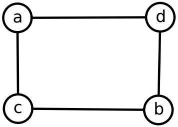
Repetimos el proceso: $(\overset{b}{1},\overset{d}{1},\overset{a}{0})$

Repetimos y terminamos: $(\overset{d}{0},\overset{a}{0})$

Grafos regulares y completos



Denominamos **grafo completo** a un grafo tal que todos los pares de vértices están conectados. Denotamos K_n al grafo completo de n vértices.

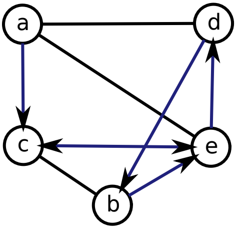


Denominamos **grafo k-regular** a todo grafo tal que todos sus vértices tienen grado k.

Todo grafo completo K_n es $n-1$ regular.

Caminos en un grafo

Camino, circuito y ciclo



acedbec
Camino de longitud 6

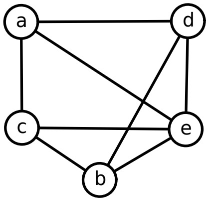
Camino: Sucesión finita de vértices de modo que cada dos consecutivos son adyacentes.

El número de aristas dentro de un camino es su **longitud**.

Se denominan **extremos** del camino al primer y último vértice.

Dos vértices u, v están **conectados** si son los extremos de algún camino.

Camino, circuito y ciclo



Un camino es **cerrado** cuando los extremos coinciden: *acbeca*

Un camino es **simple** cuando no se repite ningún vértice: *acbde*

Un **ciclo** es un camino cerrado en el que no se repite ningún otro vértice: *acbea*

Un **circuito** es un camino cerrado en el que no se repite ninguna arista: *edaebce*

Teorema

Si en un grafo G existe un camino que conecta dos vértices distintos, entonces existe un camino simple que conecta los mismos vértices.

Demostración.

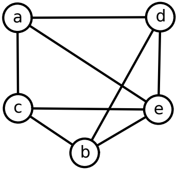
Supongamos que u, v están conectados por $u = w_0 w_1 w_2 \ldots w_n = v$. Si no es simple $w_i = w_j$, para algún $i < j$. Entonces el camino

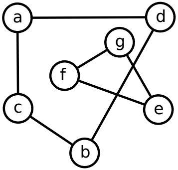
$$w_0 w_1 \ldots w_{i-1} w_i = w_j w_{j+1} \ldots w_n$$

tiene un vértice repetido menos. Repitiendo el proceso obtenemos un camino simple.

$\square$

Grafo conexo y componentes conexas





Decimos que un grafo es **conexo** si todo par de vértices están conectados.

Se denomina **componente conexa** a todo subgrafo conexo maximal (que no esté contenido en otro subgrafo conexo más grande).

En el grafo de la izquierda, hay dos componentes conexas: $\{a, b, c, d\}$ y $\{e, f, g\}$.

Conexión a partir de la matriz de adyacencia

Teorema

Sea G un grafo y M_G su matriz de adyacencia. La posición (i,j) de la matriz M_G^k es el número de caminos de longitud k que comienzan en el vértice i y terminan en el j.

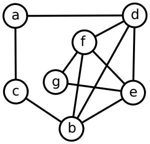
Definimos la **matriz de conexión** como

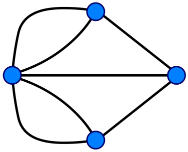
$$M_C = M_G + M_G^2 + \ldots + M_G^{n-1}$$

donde n es el número de vértices.

Entonces el grafo es conexo si y sólo si M_C no tiene ceros.

Grafo euleriano





Un grafo es **euleriano** cuando existe un circuito (camino cerrado en que no se repite ninguna arista) que recorre todas las aristas. A dicho camino lo llamamos **circuito euleriano**.

El problema de los puentes de Königsberg se reduce a encontrar un circuito euleriano.

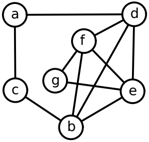
Teorema

Un grafo conexo es euleriano si y solo si todo vértice tiene grado par.

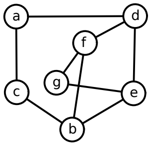
Algoritmo de Hierholzer.

1. Partimos de un vértice v cualquiera.
2. Elegimos un vértice u adyacente a v y eliminamos la arista $\{u, v\}$.
3. Repetimos el paso 2 hasta volver a v. Esto nos genera un circuito C.
4. Si no quedan aristas, hemos terminado. Si quedan aristas tomamos v un vértice de C de grado positivo.
5. Repetimos 2 y 3 y obtenemos un nuevo circuito C_1. Unimos C_1 a C.
6. Si no quedan aristas hemos terminado. En caso contrario, vamos a 4.

□



Grafos no eulerianos



¿Qué puede ocurrir si el grafo no es euleriano?
Puede existir un **camino abierto euleriano**: camino abierto que recorre todas las aristas sin repetir ninguna.

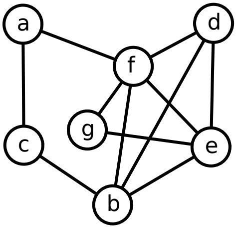
Teorema

Un grafo conexo tiene un camino abierto euleriano si y sólo si tiene exactamente 2 vértices de grado impar.

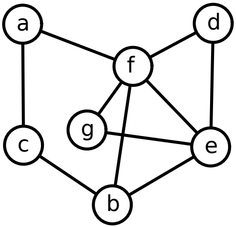
Si añadimos una arista que una los dos vértices de grado impar, así pasan a tener grado par. Creamos un circuito euleriano, y luego borramos la arista que hemos añadido.

Grafos hamiltonianos

Un grafo es **hamiltoniano** cuando existe un ciclo que recorre todos los vértices.



Hamiltoniano



No hamiltoniano

Teorema (de Ore)

Sea G un grafo con $n \geq 3$ vértices. Si cada vértice tiene grado $\geq n/2$, entonces G es hamiltoniano.

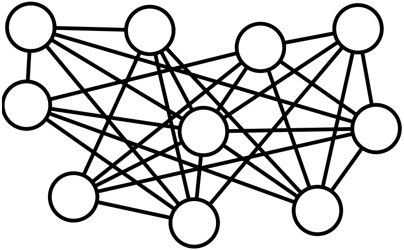
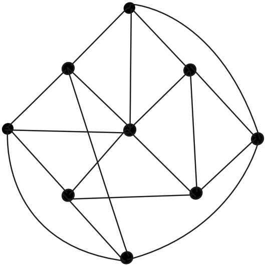
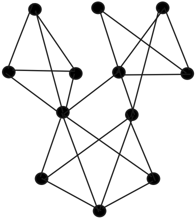


Figura: El grafo es hamiltoniano por el Teorema de Ore

Teorema (Criterio de Euler)

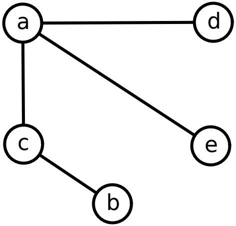
Sea G un grafo conexo con más de dos vértices. Si existe un vértice de G tal que al borrarlo junto con todas las aristas que lo tienen por extremo, se obtiene un subgrafo no conexo, entonces G no es hamiltoniano.

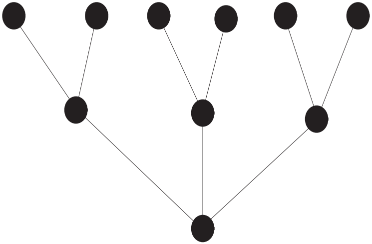




Árboles y grafos dirigidos

Decimos que un grafo es un **árbol** si es conexo y no tiene ciclos.





PROPIEDADES:

1. Un grafo T es un árbol si y solo si cada dos vértices distintos de T se conectan por un único camino simple.
2. Un grafo conexo de n vértices es un árbol si y sólo si tiene $n - 1$ aristas.

Un grafo conexo de n vértices es un árbol si y sólo si tiene $n-1$ aristas.

Demostración.

Lo probamos por inducción sobre n: se prueba que es cierto para $n=1$, y asumiendo que es cierto para todos los casos $k<n$, se prueba para n. Así se cumplirá para todo n.
Si $n=1$, está claro que un grafo de un vértice sin ciclos no puede tener aristas. Supongamos que se cumple para árboles de menos de n vértices y veamos que es cierto para un árbol de n vértices. Si eliminamos una arista del árbol, entonces pasa a tener dos componentes conexas, T_1, T_2, sin ciclos, luego dos árboles. Sea n_1 el número de vértices de T_1, y n_2 el de T_2. $n_1, n_2 < n$ y $n_1 + n_2 = n$. Por inducción T_1 tiene $n_1 - 1$ aristas y T_2 tiene $n_2 - 1$ aristas. Entonces T tiene $n_1 - 1 + n_2 - 1 + 1 = n - 1$ aristas. $\square$

Árboles

Decimos que un vértice v es **terminal** si $\mathrm{gr}(v) = 1$.

Teorema

Un árbol de $n \geq 2$ vértices tiene al menos dos vértices terminales.

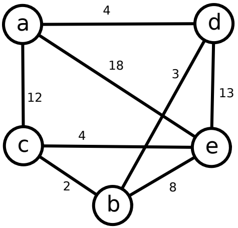
Demostración.

Supongamos que no es cierto, esto es, tiene 0 o 1 vértice terminal. Para todo árbol de n vértices, se cumple que la suma de los grados es igual al doble del número de aristas, así que

$$2(n-1) = \sum_{v} \mathrm{gr}(v) \geq 1 + 2(n-1)$$

lo cual lleva a contradicción. $\square$

Grafo etiquetado



Se denomina **grafo etiquetado** a un grafo cuyas aristas poseen etiquetas, que se pueden entender como distancias o costes.

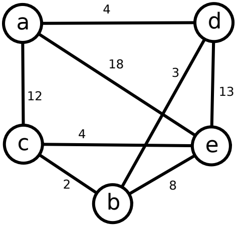
Definimos **longitud** de un camino como la suma de las etiquetas de las aristas.
La **distancia** entre dos vértices u, v, $d(u, v)$, es el mínimo de las longitudes de los caminos que unen u y v.

Algoritmo de Dijkstra

Teorema (Algoritmo de Dijkstra)

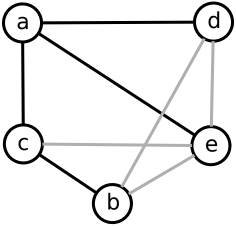
La distancia entre dos vértices x e y se puede obtener por el siguiente procedimiento:

1. *Definimos el conjunto de vértices $T = V$ y la función distancia $L(x) = 0$, $L(v) = +\infty$ para todo $v \neq x$.*
2. *Calculamos $v \in T$ que minimice la distancia de x a v. Es decir, partimos de $v = x$.*
3. *Si $v = y$, entonces hemos terminado y la distancia es $L(y)$.*
4. *En caso contrario, hacemos $T \to T - \{v\}$, $L(w) \to \min\{L(w), L(v) + d(v, w)\}$ y volvemos al paso 2.*



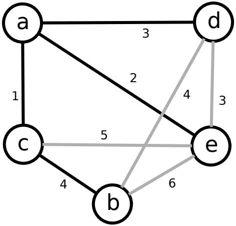
Distancia mínima de a a e.
$T = (a, b, c, d, e)$
$L = (0, \infty, \infty, \infty, \infty)$
Tomamos $v = a$:
$T = (b, c, d, e)$ $L = (\infty, 12, 4, 18)$
Tomamos $v = d$:
$T = (b, c, e)$, $L = (7, 12, 17)$
Tomamos $v = b$:
$T = (c, e)$, $L = (9, 15)$
Tomamos $v = c$:
$T = (e)$, $L = (13)$
La distancia es 13

Árbol recubridor



Dado un grafo conexo G, se denomina **árbol recubridor** o **generador** de G a cualquier árbol que sea subgrafo de G y contenga todos los vértices.

Árbol recubridor minimal



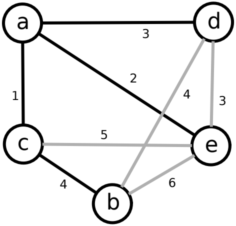
Dado un grafo etiquetado, se denomina **Árbol recubridor minimal** a cualquier árbol recubridor tal que la suma de los pesos de las aristas sea menor o igual a la de cualquier otro árbol recubridor.

Algoritmo de Kruskal

Partimos de un grafo conexo y etiquetado G.

1. Partimos del conjunto de todos los vértices y ninguna arista.
2. Elegimos la arista del grafo que tenga menor peso. La añadimos al grafo, siempre que no forme ningún ciclo.
3. Seguimos realizando este procedimiento hasta que consigamos un árbol que contenga todos los vértices.

Algoritmo de Kruskal



En E sólo indicaremos las aristas en cada paso.

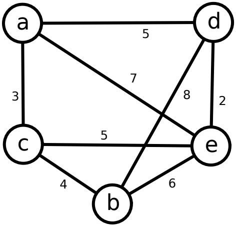
$E = \{\}$
$E = \{\{a, c\}\}$
$E = \{\{a, c\}, \{a, e\}\}$
$E = \{\{a, c\}, \{a, e\}, \{a, d\}\}$
Descartamos $\{d, e\}$ (formaría un ciclo)
$E = \{\{a, c\}, \{a, e\}, \{a, d\}, \{c, b\}\}$

Algoritmo de Prim

Partimos de un grafo conexo etiquetado.

1. Definimos un grafo T formado incialmente por uno de los vértices (escogido al azar).
2. De todos los vértices adyacentes a T que no estén en T, añadimos aquel cuya distancia sea mínima, así como dicha arista que lo conecta con T.
3. Repetimos el paso anterior hasta agotar todos los vértices.

Algoritmo de Prim



En E indicaremos las aristas y en T los vértices en cada paso.

Partimos del vértice a como único vértice de T

$T = \{a\}$, $E = \{\}$

$T = \{a, c\}$, $E = \{\{a, c\}\}$

$T = \{a, c, b\}$,$E = \{\{a, c\}, \{c, b\}\}$

$T = \{a, c, b, d\}$,

$E = \{\{a, c\}, \{c, b\}, \{a, d\}\}$

$T = \{a, c, b, d, e\}$,

$E = \{\{a, c\}, \{c, b\}, \{a, d\}, \{d, e\}\}$

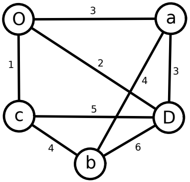
Flujo máximo

En los problemas de flujo en redes, las aristas representan canales por los que pueden circular: datos, agua, coches, corriente eléctrica...entre un origen y un destino.

El **problema del flujo máximo** consiste en:

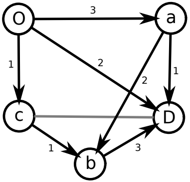
- Encontrar dentro de un grafo etiquetado un subgrafo dirigido, con la cantidad máxima de flujo que puede circular entre el origen y el destino.
- Ley de conservación de flujo: Para todo vértice, la suma de los pesos de las aristas que salen $=$ suma de los pesos de las aristas que entran
- Los pesos de las aristas del subgrafo dirigido no pueden ser mayores que las del grafo inicial.

Ejemplo flujo



$$\begin{array}{c|c} & \begin{matrix} \;O & \;\;a & \;\;b & \;\;c & \;\;D \end{matrix} \\ \begin{matrix} O \\ a \\ b \\ c \\ D \end{matrix} & \begin{pmatrix} 0 & 3 & 0 & 1 & 2 \\ -3 & 0 & 2 & 0 & 1 \\ 0 & -2 & 0 & -1 & 3 \\ -1 & 0 & 1 & 0 & 0 \\ -2 & -1 & -3 & 0 & 0 \end{pmatrix} \end{array}$$

Matriz del flujo

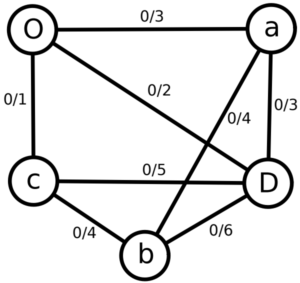


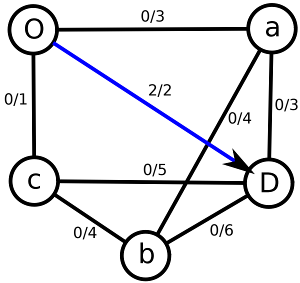
Algoritmo de Ford-Fulkenson

La **capacidad residual** de dos nodos $c(u, v)$ es la diferencia entre el peso de la arista $\{u, v\}$ de G y el flujo que circula $F(u, v)$.
Una **trayectoria de aumento** es un camino simple de O a D tal que para cada dos vértices u, v del camino, tenemos $c(u, v) > 0$.

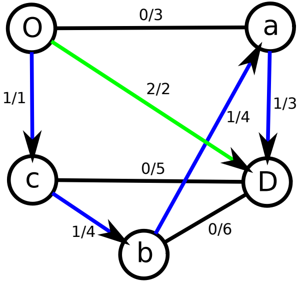
1. Comenzamos con un flujo F tal que $F(u, v) = 0$ para todo par de vértices u, v.
2. Buscamos una trayectoria de aumento que sea la de mayor capacidad. Sea $c_t > 0$ el mínimo de $c(u, v)$ donde (u, v) recorre las aristas de la trayectoria.
3. Para cada par de vértices (u, v) de la trayectoria, recalculamos $F(u, v) = F(u, v) + c_t$ y $F(v, u) = F(v, u) - c_t$.
4. Volvemos a 2 hasta que no encontremos más trayectorias de aumento. El flujo máximo será la suma de todos los flujos que salen del origen=suma de los que llegan al destino.

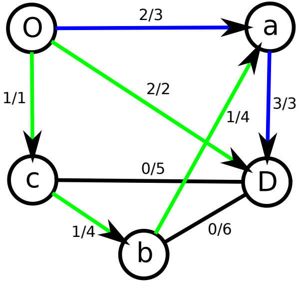
Algoritmo de Ford-Fulkenson



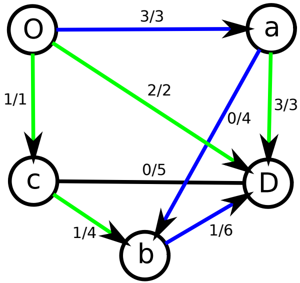


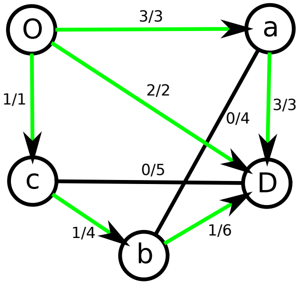
Algoritmo de Ford-Fulkenson



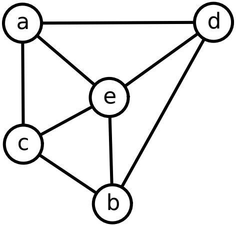


Algoritmo de Ford-Fulkenson





Mapas y coloraciones



Un grafo es **plano** si admite una representación gráfica (**mapa**) en el plano de modo que no se corten las aristas.

Dado un mapa, se denomina **región** a un camino cerrado tal que en su "interior" no hay ni vértices ni aristas.
Dos regiones son **adyacentes** si tienen una arista en común.

Se denomina **grado de una región** a la longitud del camino que la rodea (número de aristas que la delimitan).
(nótese que lo de "fuera" del grafo es también una región)

Teorema

Sea G un grafo plano y R el conjunto de regiones de un mapa asociado. Entonces

$$\sum_{r\in R} gr(r_i) = 2card(E).$$

Teorema (Fórmula de Euler)

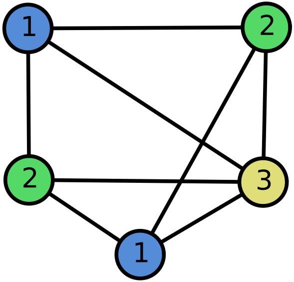
Sea M un mapa conexo, que representa al grafo $G = (V, E)$. Entonces

$$card(V) - card(E) + card(R) = 2.$$

Como consecuencia:
Si un grafo es conexo y posee al menos tres vértices, si es plano entonces

$$card(E) \leq 3(card(V) - 2)$$

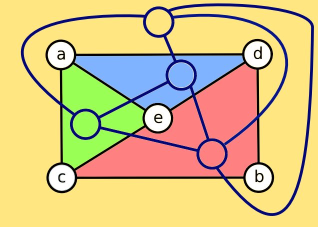
Por ejemplo, K_5 no es plano.



Dada una paleta de colores C, una **coloración por vértices** de G con k colores es una aplicación $\gamma\colon V \to C$ de modo que si $u, v \in V$ son vértices adyacentes, entonces $\gamma(u) \neq \gamma(v)$.

Una **coloración de las regiones** de un mapa es una aplicación $\gamma\colon R \to C$ de modo que si $r, s \in R$ son regiones adyacentes, entonces $\gamma(r) \neq \gamma(s)$.

Teorema de los cuatro colores



Teorema (K. Appel, W. Haken, J. Koch)

Todo grafo plano admite una coloración con a lo sumo cuatro colores.

Corolario

Todo mapa admite una coloración de las regiones con a lo sumo cuatro colores de modo que dos regiones adyacentes tengan distinto color.