

Teoría de grafos

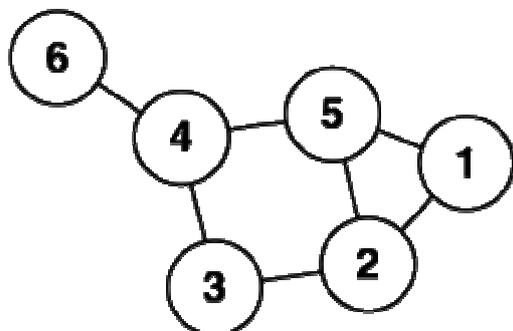


 Diagrama de un grafo con 6 vértices y 7 aristas.

En [matemáticas](#) y [ciencias de la computación](#), la **teoría de grafos** estudia las propiedades de los [grafos](#), que son colecciones de objetos llamados [vértices](#) (o nodos) conectados por líneas llamadas [aristas](#) (o arcos) que pueden tener orientación (dirección asignada). Típicamente, un grafo está diseñado por una serie de puntos (los vértices) conectados por líneas (las aristas).

Tabla de contenidos

- [1 Historia](#)
- [2 Definiciones](#)
 - [2.1 Grafo](#)
 - [2.2 Aristas dirigidas y no dirigidas](#)
 - [2.3 Ciclos y caminos hamiltonianos](#)
 - [2.4 Caracterización de Grafos](#)
 - [2.4.1 Grafos Simples](#)
 - [2.4.2 Grafos Conexos](#)
 - [2.4.3 Grafos Completos](#)
 - [2.4.4 Grafos Bipartitos](#)
 - [2.5 Árboles](#)
 - [2.6 Grafos ponderados](#)
 - [2.7 Teorema de los cuatro colores](#)
 - [2.8 Coloración de Grafos](#)
 - [2.9 Grafos planos](#)
 - [2.10 Diámetro](#)
- [3 Algoritmos importantes](#)
- [4 Aplicaciones](#)
- [5 Investigadores relevantes en Teoría de grafos](#)
- [6 Véase también](#)
- [7 Enlaces externos](#)

[\[editar\]](#)

Historia

El trabajo de [Leonhard Euler](#), en [1736](#), sobre el [problema de los puentes de Königsberg](#) es considerado como uno de los primeros resultados de la teoría de grafos. También se considera uno de los primeros resultados topológicos en geometría (que no depende de ninguna medida). Este ejemplo ilustra la profunda relación entre la teoría de grafos y la [topología](#).

En [1845](#) [Gustav Kirchhoff](#) publicó sus leyes de los circuitos para calcular el voltaje y la corriente en los circuitos eléctricos.

En [1852](#) [Francis Guthrie](#) planteó el [problema de los cuatro colores](#) que plantea si es posible, utilizando solamente cuatro colores, colorear cualquier mapa de países de tal forma que dos países vecinos nunca tengan el mismo color. Este problema, que no fue resuelto hasta un siglo después por [Kenneth Appel](#) y [Wolfgang Haken](#), puede ser considerado como el nacimiento de la teoría de grafos. Al tratar de resolverlo, los matemáticos definieron términos y conceptos teóricos fundamentales de los grafos.

[[editar](#)]

Definiciones

[[editar](#)]

Grafo

Artículo principal: [Grafo](#)



En la figura, $V = \{ a, b, c, d, e, f \}$, y $A = \{ ab, ac, ae, bc, bd, df, ef \}$.

Un grafo es una pareja $G = (V, A)$, donde V es un conjunto de puntos, llamados vértices, y A es un conjunto de pares de vértices, llamadas aristas. Para simplificar, notaremos la arista $\{a, b\}$ como ab .

En teoría de grafos, sólo queda lo esencial del dibujo: la forma de las aristas no son relevantes, sólo importa a qué vértices están unidas. La posición de los vértices tampoco importa, y se puede variar para obtener un grafo más claro. Generalmente, se considera que colocar los vértices en forma de polígono regular da grafos muy legibles.

Prácticamente cualquier red puede ser modelada con un grafo: una red de carreteras que conecta ciudades, una red eléctrica o un alcantarillado.

[[editar](#)]

Aristas dirigidas y no dirigidas



En algunos casos es necesario asignar un sentido a las aristas, por ejemplo, si se quiere representar la red de las calles de una ciudad con sus inevitables direcciones únicas. El conjunto de aristas será ahora un subconjunto de todos los posibles pares ordenados de vértices, con $(a, b) \neq (b, a)$. Los grafos que contienen aristas dirigidas se denominan **grafos orientados**, como el siguiente:

Las aristas no orientadas se consideran bidireccionales para efectos prácticos (equivale a decir que existen dos aristas orientadas entre los nodos, cada una en un sentido).

En el grafo anterior se ha utilizado una arista que tiene sus dos extremos idénticos: es un lazo (o bucle), y aparece también una arista bidireccional, y corresponde a dos aristas orientadas.

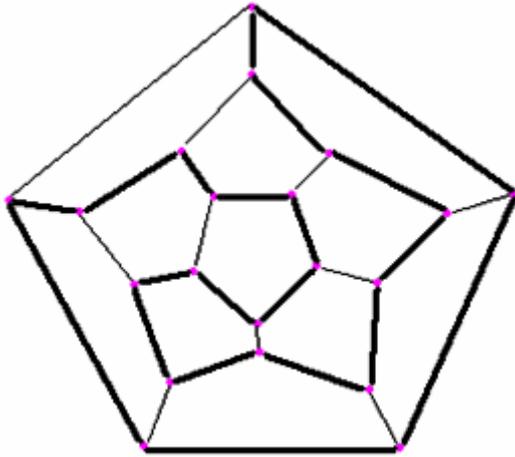
Aquí $V = \{ a, b, c, d, e \}$, y $A = \{ (a, c), (d, a), (d, e), (a, e), (b, e), (c, a), (c, c), (d, b) \}$.

Se considera la característica de "grado" (positivo o negativo) de un vértice, como la cantidad de aristas que llegan o salen de él; para el caso de grafos no orientados, el grado de un vértice es simplemente la cantidad de aristas que tocan este vértice. Por ejemplo, el grado positivo (salidas) de d es 3, mientras que el grado negativo (llegadas) de b es 1.

Según la terminología seguida en algunos problemas clásicos de [Investigación Operativa](#) (p.ej.: el [Problema del flujo máximo](#)), a un vértice del que sólo salen aristas se le denomina *fuentes* (en el ejemplo anterior, el vértice d); tiene grado negativo 0. Por el contrario, a aquellos en los que sólo entran aristas se les denomina *pozos* o *sumideros* (en el caso anterior, el vértice e); tiene grado positivo 0.

[\[editar\]](#)

Ciclos y caminos hamiltonianos



Ejemplo de un camino hamiltoniano.

Un **ciclo** es un camino, es decir una sucesión de aristas adyacentes, donde no se recorre dos veces la misma arista, y donde se regresa al punto inicial. Un **ciclo hamiltoniano** tiene además que recorrer todos los vértices.

Por ejemplo, en un museo grande (al estilo del [Louvre](#)), lo idóneo sería recorrer todas las salas una sola vez, esto es buscar un ciclo hamiltoniano en el grafo que representa el museo (los vértices son las salas, y las aristas los corredores o puertas entre ellas).

Se habla también de [camino hamiltoniano](#) si no se impone regresar al punto de partida, como en un museo con una única puerta de entrada. Por ejemplo, un caballo puede recorrer todas las casillas de un tablero de ajedrez sin pasar dos veces por la misma: es un camino hamiltoniano. Ejemplo de un ciclo hamiltoniano en el grafo del dodecaedro.

Hoy en día, no se conocen métodos generales para hallar un ciclo hamiltoniano en [tiempo polinómico](#), siendo la búsqueda por fuerza bruta de todos los posibles caminos u otros métodos excesivamente costosos. Existen, sin embargo, métodos para descartar la existencia de ciclos o caminos hamiltonianos en grafos pequeños.

El problema de determinar la existencia de ciclos hamiltonianos, entra en el conjunto de los [NP-completos](#).

[\[editar\]](#)

Caracterización de Grafos

[\[editar\]](#)

Grafos Simples

Un grafo es *simple* si a lo más *sólo* 1 arista une dos vértices cualesquiera. Esto es equivalente a decir que una arista cualquiera es el único que une dos vértices específicos.

Un grafo que *no* es simple se denomina *complejo*.

[\[editar\]](#)

Grafos Conexos

Un grafo es conexo (más formalmente *fuertemente conexo*) si todos sus vértices están conectados por un camino; es decir, si para cualquier par de vértices (a, b) , existe al menos un camino posible desde a hacia b .

Es posible determinar si un grafo es fuertemente conexo coleccionando la información de los grados de sus vértices al tiempo que se acumulan las diferentes rutas que salen de un vértice o llegan a él.

En términos matemáticos la propiedad de un grafo de ser fuertemente conexo permite establecer en base a él una [relación de equivalencia](#) para sus vértices, la cual lleva a una partición de éstos en "componentes fuertemente conexos", es decir, porciones del grafo, que son fuertemente conexas cuando se consideran como grafos aislados. Esta propiedad es importante para muchas demostraciones en teoría de grafos.

[\[editar\]](#)

Grafos Completos

Un grafo simple es *completo* si existen aristas uniendo *todos* los pares posibles de vértices. Es decir, todo par de vértices (a, b) debe tener una arista e que los une.

El conjunto de los grafos completos es denominado usualmente \mathbb{K} , siendo \mathbb{K}_n el grafo completo de n vértices.

Un \mathbb{K}_n , es decir, grafo completo de n vértices tiene exactamente
$$\frac{n \times (n - 1)}{2}$$
 aristas.

La representación gráfica de los \mathbb{K}_n como los vértices de un polígono regular da cuenta de su peculiar estructura.

[\[editar\]](#)

Grafos Bipartitos

Artículo principal: [Grafo bipartito](#)

Un grafo G es bipartito si puede expresarse como $G = \{V_1 + V_2, A\}$ (es decir, la unión de dos grupos de vértices), bajo las siguientes condiciones:

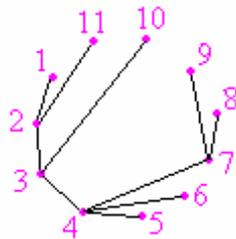
- V_1 y V_2 son distintos y tienen más de un elemento cada uno.
- Una arista en A une un vértice de V_1 con uno de V_2 .
- No existen aristas uniendo dos elementos de V_1 ; análogamente para V_2 .

Bajo estas condiciones, el grafo se considera bipartito, y puede describirse informalmente como el grafo que une o relaciona dos conjuntos de elementos

diferentes, como aquellos resultantes de los ejercicios y puzzles en los que debe unirse un elemento de la columna A con un elemento de la columna B.

[[editar](#)]

Árboles



Ejemplo de árbol.

Un grafo que no tiene ciclos y que conecta a todos los puntos, se llama un **árbol**. En un grafo con n vértices, los árboles tienen exactamente $n - 1$ aristas, y hay n^{n-2} árboles posibles. Su importancia radica en que los árboles son grafos que conectan vértices utilizando el menor número posible de aristas. Un importante campo de aplicación de su estudio se encuentra en el [análisis filogenético](#), el de la filiación de entidades que derivan unas de otras en un proceso evolutivo, que se aplica sobre todo a la averiguación del parentesco entre especies; aunque se ha usado también, por ejemplo, en el estudio del parentesco entre lenguas.

[[editar](#)]

Grafos ponderados

En muchos casos, es preciso atribuir a cada arista un número específico, llamado *valuación*, *ponderación* o *coste* según el contexto, y se obtiene así un **grafo valuado**. Formalmente, es un grafo con una función $v: A \rightarrow \mathbf{R}_+$.

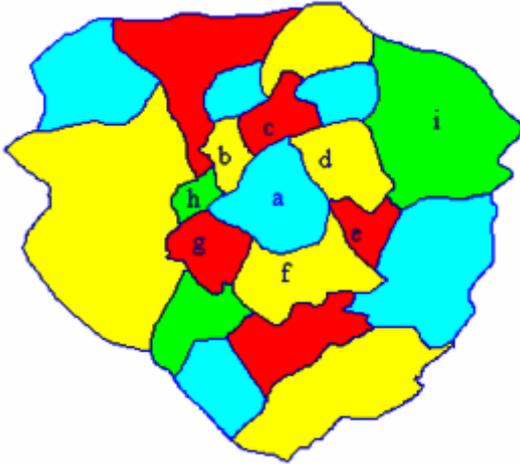
Por ejemplo, un representante comercial tiene que visitar n ciudades conectadas entre sí por carreteras; su interés previsible será minimizar la distancia recorrida (o el tiempo, si se pueden prever atascos). El grafo correspondiente tendrá como vértices las ciudades, como aristas las carreteras y la valuación será la distancia entre ellas.

Y, de momento, no se conocen métodos generales para hallar un ciclo de valuación mínima, pero sí para los caminos desde a hasta b , sin más condición.

[[editar](#)]

Teorema de los cuatro colores

Artículo principal: [Teorema de los cuatro colores](#)



En 1852 Francis Guthrie planteó el problema de los cuatro colores.

Otro problema famoso relativo a los grafos: ¿Cuántos colores son necesarios para dibujar un mapa político, con la condición obvia que dos países adyacentes no puedan tener el mismo color? Se supone que los países son de un solo pedazo, y que el mundo es esférico o plano. En un mundo en forma de toro; el teorema siguiente no es válido:

Cuatro colores son siempre suficientes para colorear un mapa.

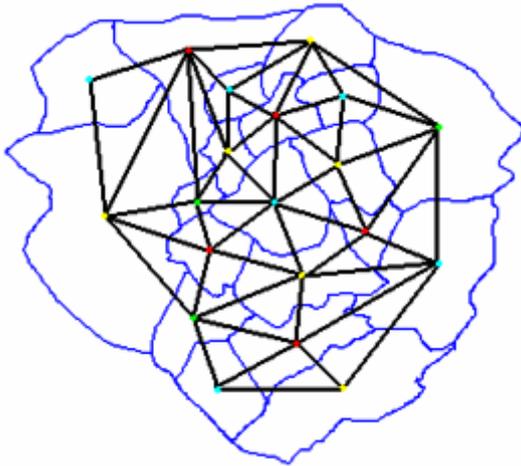
El mapa siguiente muestra que tres colores no bastan: Si se empieza por el país central **a** y se esfuerza uno en utilizar el menor número de colores, entonces en la corona alrededor de **a** alternan dos colores. Llegando al país **h** se tiene que introducir un cuarto color. Lo mismo sucede en **i** si se emplea el mismo método.

La forma precisa de cada país no importa; lo único relevante es saber qué país toca a qué otro. Estos datos están incluidos en el grafo donde los vértices son los países y las aristas conectan los que justamente son adyacentes. Entonces la cuestión equivale a atribuir a cada vértice un color distinto del de sus vecinos.

Hemos visto que tres colores no son suficientes, y demostrar que con cinco siempre se llega, es bastante fácil. Pero el teorema de los cuatro colores no es nada obvio. prueba de ello es que se ha tenido que emplear los ordenadores para acabar la demostración (se ha hecho un programa que permitió verificar una multitud de casos, lo que ahorró muchísimo tiempo a los matemáticos). Fue la primera vez que la comunidad matemática aceptó una demostración asistida por ordenador.

[\[editar\]](#)

Coloración de Grafos



Colores en los vértices.

Definición: Si $G=(V,E)$ es un grafo no dirigido, una coloración propia de G , ocurre cuando coloreamos los vértices de G de modo que si $\{a, b\}$ es una arista en G entonces a y b tienen diferentes colores. (Por lo tanto, los vértices adyacentes tienen colores diferentes). El número mínimo de colores necesarios para una coloración propia de G es el número cromático de G y se escribe como $C(G)$. Sea G un grafo no dirigido sea λ el número de colores disponibles para la coloración propia de los vértices de G . Nuestro objetivo es encontrar una función polinomial $P(G,\lambda)$, en la variable λ , llamada polinomio cromático de G , que nos indique el número de coloraciones propias diferentes de los vértices de G , usando un máximo de λ colores.

Descomposición de polinomios cromáticos. Si $G=(V,E)$ es un grafo conexo y e pertenece a E , entonces: $P(G,e,\lambda)=P(G,\lambda)+P(G-e,\lambda)$

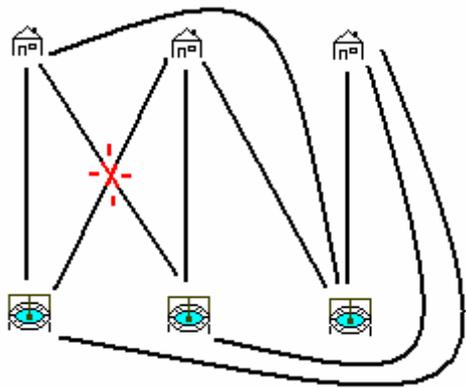
Para cualquier grafo G , el término constante en $P(G,\lambda)$ es 0

Sea $G=(V,E)$ con $|E|>0$ entonces, la suma de los coeficientes de $P(G,\lambda)$ es 0.

Sea $G=(V,E)$, con a, b pertenecientes al conjunto de vértices V pero $\{a, b\}=e$ no perteneciente a al conjunto de aristas E . Escribimos $G+e$ para el grafo que se obtiene de G al añadir la arista $e=\{a, b\}$. Al identificar los vértices a y b en G , obtenemos el subgrafo $G++e$ de G . (Leonardo Sandoval)

[\[editar\]](#)

Grafos planos



Un grafo es **plano** si se puede dibujar sin cruces de aristas.

Cuando un grafo o multigrafo se puede dibujar en un plano sin que dos segmentos se corten, se dice que es plano.

Un juego muy conocido es el siguiente: Se dibujan tres casas y tres pozos. Todos los vecinos de las casas tienen el derecho de utilizar los tres pozos. Como no se llevan bien en absoluto, no quieren cruzarse jamás. ¿Es posible trazar los nueve caminos que juntan las tres casas con los tres pozos sin que haya cruces?

Cualquier disposición de las casas, los pozos y los caminos implica la presencia de al menos un cruce.

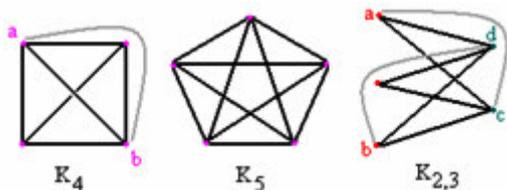
Sea K_n el **grafo completo** con n vértices, $K_{n,p}$ es el grafo bipartito de n y p vértices.

El juego anterior equivale a descubrir si el grafo $K_{3,3}$, bipartito completo, es **planario** (se dice también **plano**), es decir, si se puede dibujar en un plano sin que haya cruces. Y la respuesta es no. En general, puede determinarse que un grafo *no* es plano, si en su diseño puede encontrarse una estructura análoga a K_5 o a $K_{3,3}$.

Establecer qué grafos son planos no es obvio, y tiene que ver con la [topología](#).

[\[editar\]](#)

Diámetro



En la figura se nota que K_4 es plano (desviando la arista ab al exterior del cuadrado), que K_5 no lo es, y que $K_{3,2}$ lo es también (desvíos en gris).

En un grafo, la distancia entre dos vértices es el menor número de aristas de un recorrido entre ellos. El **diámetro**, en una figura como en un grafo, es la mayor distancia entre dos puntos de la misma.

El diámetro de los K_n es 1, y el de los $K_{n,p}$ es 2. Un diámetro infinito puede significar que el grafo tiene una infinidad de vértices o simplemente que no es **conexo**. También se puede considerar el **diámetro promedio**, como el promedio de las distancias entre dos vértices.

El mundo de Internet ha puesto de moda esa idea del diámetro: Si descartamos los sitios que no tienen enlaces, y escogemos dos páginas *web* al [azar](#): ¿En cuántos *clicks* se puede pasar de la primera a la segunda? El resultado es el diámetro de la Red, vista como un grafo cuyos vértices son los sitios, y cuyas aristas son lógicamente los enlaces.

En el mundo real hay una analogía: tomando al azar dos seres humanos del mundo, ¿En cuántos saltos se puede pasar de uno a otro, con la condición de sólo saltar de una persona a otra cuando ellas se conocen personalmente? Con esta definición, se estima que el diámetro de la humanidad es de... ¡ocho solamente!

Este concepto refleja mejor la complejidad de una red que el número de sus elementos.

{ {MI}[[Glosario en teoría de grafos] }

[\[editar\]](#)

Algoritmos importantes

- [Algoritmo de búsqueda en anchura \(BFS\)](#)
- [Algoritmo de búsqueda en profundidad \(DFS\)](#)
- [Ordenación topológica de un grafo](#)
- [Algoritmo de cálculo de los componentes fuertemente conexos de un grafo](#)
- [Algoritmo de Dijkstra](#)
- [Algoritmo de Bellman-Ford](#)
- [Algoritmo de Prim](#)
- [Algoritmo de Ford-Fulkerson](#)
- [Algoritmo de Kruskal](#)
- [Algoritmo del vecino más cercano](#)

[\[editar\]](#)

Aplicaciones

Gracias a la teoría de Grafos se pueden resolver diversos problemas como por ejemplo la síntesis de [circuitos](#) secuenciales, contadores o sistemas de apertura.

Los grafos se utilizan también para modelar trayectos como el de una línea de autobús a través de las calles de una ciudad, en el que podemos obtener caminos óptimos para el trayecto aplicando diversos [algoritmos](#) como puede ser el algoritmo de [Floyd](#).

Para la administración de proyectos, utilizamos técnicas como [PERT](#) en las que se modelan los mismos utilizando grafos y optimizando los tiempos para concretar los mismos.

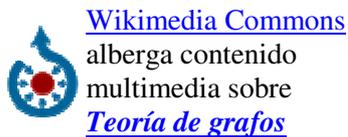
[[editar](#)]

Investigadores relevantes en Teoría de grafos

- [Paul Erdős](#)
- [Frank Harary](#)
- [Denes König](#)
- [W.T. Tutte](#)

[[editar](#)]

Véase también



- [Grafo](#)
- [Modelo en grafo](#)
- [Algoritmo de Floyd](#)

[[editar](#)]

Enlaces externos

- [páginas blancas de la Teoría de grafos](#) (para más investigadores y publicaciones).
- ["Teoría de grafos" en la Enciclopedia Libre Universal en Español](#)
- **Grafos**: Es un [software](#) para la construcción, edición y análisis de grafos. Forma parte de un proyecto ([copyleft](#)) de investigación y desarrollo de aplicaciones informáticas de diseño modular orientadas hacia la docencia, investigación y labores profesionales de [Ingeniería de Organización Industrial](#) ([Management Engineering](#)).

Fuente: http://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos

Programación no numérica - Grafos

Definición de grafo:

Desafortunadamente no existe una terminología estandarizada en la **teoría** de los **grafos**, por lo tanto es oportuno aclarar que las presentes definiciones pueden variar ligeramente entre diferentes publicaciones de **estructura de datos** y de **teoría de grafos**, pero en general se puede decir que un grafo como indica su nombre lo indica es la representación (para nuestro caso) gráfica de los **datos** de una situación particular, ejemplo:

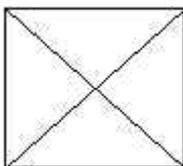
Los **datos** contienen, en algunos casos, relaciones entre ellos que no es necesariamente jerárquica. Por ejemplo, supongamos que unas líneas aéreas realizan vuelos entre las ciudades conectadas por líneas como se ve en la figura anterior (más adelante se presentaran **grafos** con **estructuras de datos**); la **estructura de datos** que refleja esta relación recibe el nombre de grafo.

Se suelen usar muchos nombres al referirnos a los elementos de una **estructura de datos**. Algunos de ellos son "elemento", "ítem", "asociación de ítems", "**registro**", "nodo" y "objeto". El nombre que se utiliza depende del tipo de **estructura**, el contexto en que usamos esa **estructura** y quien la utiliza.

En la mayoría de los textos de **estructura de datos** se utiliza el termino "**registro**" al hacer referencia a **archivos** y "nodo" cuando se usan listas enlazadas, **arboles** y **grafos**.

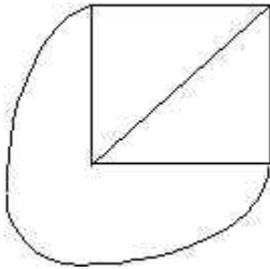
También un grafo es una terna $G = (V, A, j)$, en donde **V** y **A** son **conjuntos** finitos, y **j** es una aplicación que hace corresponder a cada elemento de **A** un par de elementos de **V**. Los elementos de **V** y de **A** se llaman, respectivamente, "**vértices**" y "**aristas**" de **G**, y **j** asocia entonces a cada arista con sus dos vértices.

Esta definición da lugar a una representación gráfica, en donde cada vértice es un punto del plano, y cada arista es una línea que une a sus dos vértices.



Si el **dibujo** puede efectuarse sin que haya superposición de líneas, se dice que **G** es un grafo plano. Por ejemplo, el siguiente es un grafo plano:

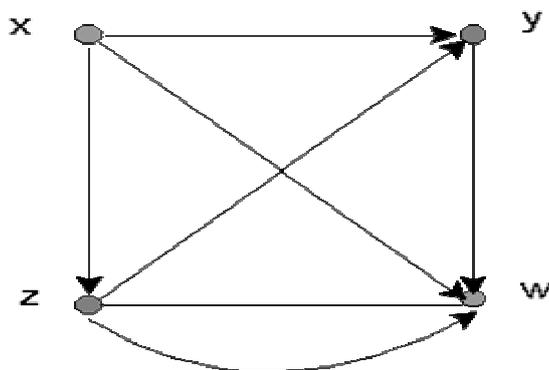
puesto que es equivalente a este otro:



Representación de un grafo:

Existen dos formas de mantener un grafo "**G**" en la memoria de una computadora, una se llama **Representación secuencial** de **G**, la cual se basa en la matriz de adyacencia **A**; la otra forma, es la llamada **Representación enlazada** de **G** y se basa en listas enlazadas de vecinos. Independientemente de la forma en que se mantenga un grafo **G** en la memoria de una computadora, el grafo **G** normalmente se introduce en la computadora por su definición formal: Un conjunto de nodos y un conjunto de aristas

- **Representación secuencial de un grafo:**



Considere el grafo siguiente "**G**":

y suponga que los nodos se mantienen en memoria en un array **DATOS** tal como sigue:

DATOS: X, Y, Z, W

Para hallar la matriz de adyacencia **A** del grafo "**G**", tenemos que tomar en cuenta que los nodos están normalmente ordenados de acuerdo con la forma en que aparecen en memoria; o sea, asumimos que $u_1 = X$, $u_2 = Y$, $u_3 = Z$, y $u_4 = W$, la matriz de adyacencia **A** de **G** sería la siguiente:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

aquí a $i j = 1$ si hay una arista u i a $u j$; si no a $i j = 0$.

Así entonces para hallar la **matriz** de camino **P** de **G** mediante las potencias de la **matriz** de adyacencia **A**, como **G** tiene cuatro nodos se calcula A^2, A^3, A^4 , y $B_4 = A + A^2 + A^3 + A^4$:

$$A^2 = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad A^3 = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$A^4 = \begin{pmatrix} 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix} \quad B_4 = \begin{pmatrix} 0 & 5 & 6 & 8 \\ 0 & 1 & 2 & 3 \\ 0 & 3 & 3 & 5 \\ 0 & 2 & 3 & 5 \end{pmatrix}$$

por lo tanto la matriz de caminos **P** se obtiene ahora haciendo $p_{ij} = 1$ siempre que haya una entrada positiva en la matriz **B4** . así

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

La matriz de caminos **muestra** que no hay camino de u_1 a u_2 de hecho, no hay camino de ningún nodo a u_1 por tanto, **G** no es fuertemente conexo.

- **Representación enlazada de un grafo:**

Un grafo "**G**" se guarda en **memoria** como sigue:

NODO	A	B		E		D	C	
SIG	7	4	0	6	8	0	2	3
ADY	1	2		5		7	9	
	1	2	3	4	5	6	7	8

PRINCIPIO = 1, NDISP = 5

DEST	2	6	4		6	7	4		4	6
ENL	10	3	6	0	0	0	0	4	0	0
	1	2	3	4	5	6	7	8	9	10

ADISP = 8

Para dibujar el respectivo grafo "G", primero debemos buscar todos los vecinos de cada **NODO[K]** recorriendo su lista de adyacencia que tiene el puntero de adyacencia **ADY[J]**. Esto da como resultado:

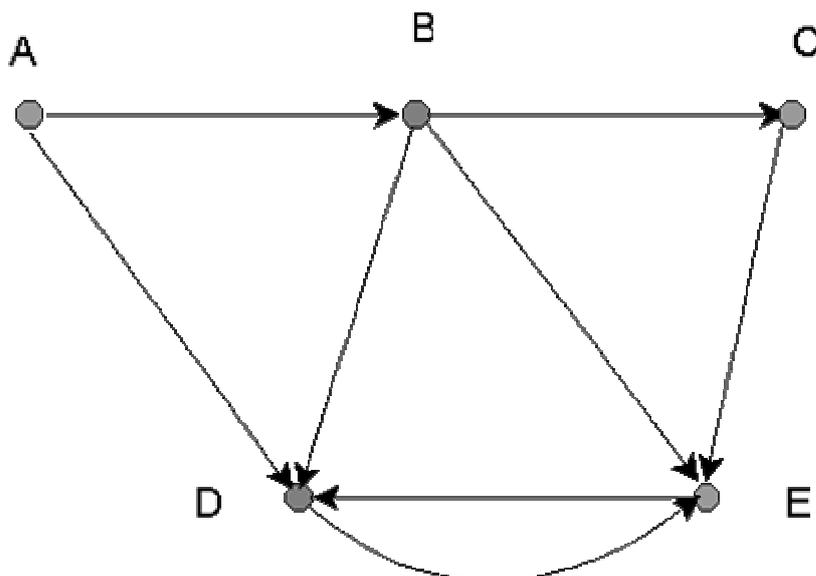
A: 2(B) y 6(D)

B: 6(D), 4(E) y 7(C)

C: 4(E)

D: 4(E)

E: 6(D)



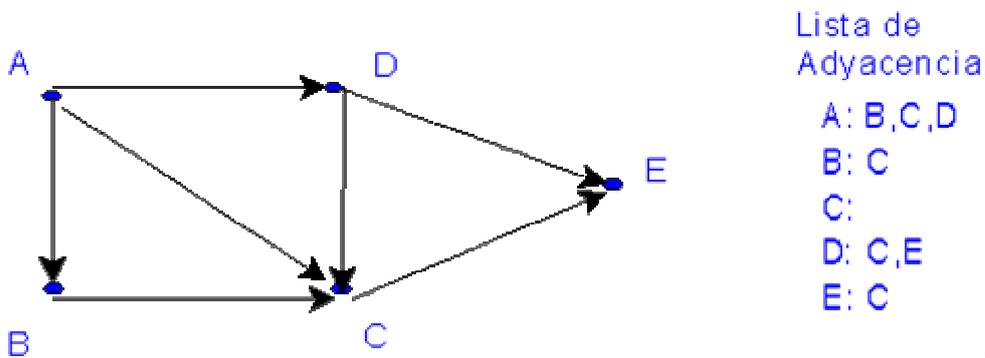
Entonces

procedemos a dibujar el **diagrama** del grafo como sigue:

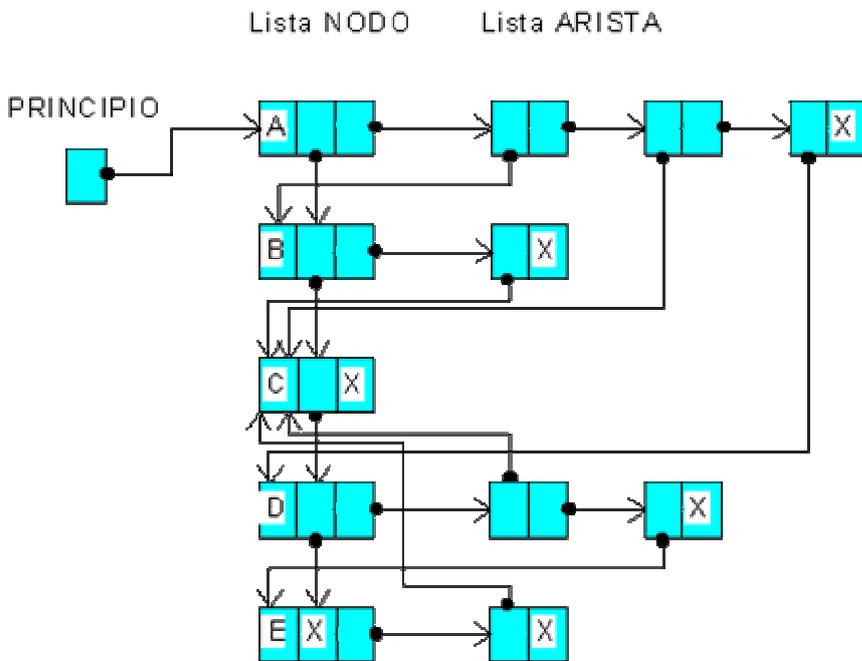
Sea **G** un grafo dirigido con m nodos. La representación secuencial de **G** en la **memoria**, o sea, la representación de **G** por su matriz de adyacencia **A**, tiene unas cuantas desventajas importantes.

En primer lugar, puede ser difícil insertar y eliminar nodos de **G**, esto es por que el tamaño de **A** debería ser cambiado y los nodos deberían ser reordenados, así que habría muchos cambios en la matriz **A**; más aun, si el numero de aristas es $O(m)$ o $O(m \log^2 m)$, entonces la matriz **A** estará

desperdiciada (contendrá muchos ceros); por tanto, se desperdiciará una gran cantidad de espacio; entonces **G** normalmente se representa en **memoria** por una representación enlazada, también llamada **estructura de adyacencia**.



Considere el grafo **G** de la figura siguiente y su respectiva tabla de adyacencia, donde se **muestra** cada nodo de **G** seguido por la lista de nodos adyacentes, también llamados **sucesores** o **vecinos**.



Para apreciar aun más esta situación, podemos también usar un **diagrama** esquemático de la representación enlazada de **G** en **la memoria**, específicamente, la representación enlazada contendrá dos listas (o **archivos**), una lista de nodos **NODO** y una lista de aristas **ARISTA**, tal como sigue:

Cada elemento de la **lista NODO** corresponderá a un nodo de **G** y será un **registro** de la forma:

NODO	SIG	ADY	
------	-----	-----	--

Aquí **NODO** será el nombre o **valor** clave del nodo, **SIG** será un puntero al siguiente nodo de la lista **NODO** y **ADY** será un puntero al primer elemento de la lista de adyacencia del nodo, que se mantiene en la **lista ARISTA**; el área

restante indica que puede haber otra **información** en el **registro**, tal como el grado de entrada **GraEnt** del nodo, el grado de salida **GraSal** del nodo, el **ESTADO** del nodo durante la ejecución de un **algoritmo**, etc.

Adicional a esto, cada elemento de la **lista ARISTA** corresponderá a una arista de **G** y será un **registro** de la forma:

DEST	ENL	
------	-----	--

Donde el campo **DEST** apuntará a la posición en la **lista NODO** del nodo destino o terminal de la arista, el campo **ENL** enlazará juntas las aristas con el mismo nodo inicial, o sea, los nodos con la misma lista de adyacencia, y el campo restante indica que puede existir otra **información** en el registro correspondiente a la arista, tal como un campo **ARIS** conteniendo los **datos** etiquetados de la arista cuando **G** es un grafo con etiquetas, un campo **PESO** conteniendo el peso de la arista cuando **G** es un grafo con peso, etc.

Técnicas básicas de búsqueda:

BÚSQUEDA EN GRAFOS

Para efectuar una búsqueda de los vértices de un grafo, se pueden emplear dos **estrategias** diferentes:

Búsqueda en profundidad (BEP): Se comienza en cualquier vértice y en cada paso se avanza a un nuevo vértice adyacente siempre que se pueda. Cuando todos los adyacentes a **X** hayan sido visitados, se retrocede al vértice desde el que se alcanzó **X** y se prosigue. Así se consigue etiquetar (visitar) todos los vértices de la componente conexa en que se encuentre el vértice inicial.

Esta técnica se utiliza cuando necesitamos encontrar respuesta a un problema sobre un grafo sin condiciones de optimización.

La idea en general de la búsqueda en profundidad comenzando en un nodo **A** es la siguiente:

Primero examinamos el nodo inicial **A**. Luego examinamos cada nodo **N** de un camino **P** que comience en **A**; a sea, procesamos un vecino de **A**, luego un vecino de un vecino de **A** y así sucesivamente, hasta llegar a un punto muerto o final del camino **P**, y de aquí volvemos atrás por **P** hasta que podamos continuar por otro camino **P'** y así sucesivamente.

Este **algoritmo** es similar al del recorrido inorden de un árbol binario, y también a la forma en que se debe pasar a través de un laberinto. Observe que se hace

uso una pila en lugar de una cola, y este es el detalle fundamental que hace la diferencia para realizar la búsqueda en profundidad.

Algoritmo para la búsqueda en profundidad:

Este algoritmo realiza la búsqueda en profundidad el grafo **G** comenzando en un nodo **A**.

1. Inicializar todos los nodos al estado de preparado (**ESTADO=1**)
 2. Meter el nodo inicial **A** en la pila y cambiar su estado a estado de espera (**ESTADO=2**).
 3. Repetir los pasos 4 y 5 hasta que la pila este vacia.
 4. Sacar el nodo **N** en la cima de la pila. Procesar el nodo **N** y cambiar su estado al de procesado (**ESTADO=3**).
 5. Meter en la pila todos los vecinos de **N** que estén en estado de preparados (**ESTADO=1**) y cambiar su estado a estado de espera (**ESTADO=2**).
- [fin de bucle del paso 3]
6. Salir.

nota: tomado del libro Estructura de datos, serie schaum Mcgraw-Hill,

pagina: 337, capitulo: 8 Grafos y sus aplicaciones, autor: Seymour Lipschutz

Búsqueda en anchura (BEA): A diferencia con la BEP ahora se visitan todos los vecinos de un vértice antes de pasar al siguiente. Por tanto no hay necesidad de retroceder. Una vez etiquetados todos los vecinos de un vértice **X**, se continúa con el primer vértice alcanzado después de **X** en la búsqueda.

Esta técnica se utiliza para resolver problemas en los que se pide hallar una solución óptima entre varias.

En general la búsqueda en anchura comenzando de un nodo de partida **A** es la siguiente:

Primero examinamos el nodo de partida **A**.

Luego examinamos todos los vecinos de **A**. Luego examinamos todos los vecinos de los vecinos de **A** y así sucesivamente. Con el uso de una cola, garantizamos que ningún nodo sea procesado más de una vez y usando un campo **ESTADO** que nos indica el estado actual de los nodos.

Algoritmo para la búsqueda en anchura:

Este algoritmo realiza la búsqueda en anchura en un grafo **G** comenzando en un nodo de partida **A**.

1. Inicializar todos los nodos al estado de preparados (ESTADO=1).
 2. Poner el nodo de partida **A** en la COLA y cambiar su estado a espera (ESTADO=2).
 3. Repetir pasos 4 y 5 hasta que COLA esté vacía.
 4. Quitar el nodo del principio de la cola, **N**. Procesar **N** y cambiar su estado a procesado (ESTADO=3).
 5. Añadir a COLA todos los vecinos de **N** que estén en estado de preparados (ESTADO=1) y cambiar su estado al de espera (ESTADO=2).
- [fin del bucle del paso 3]
6. Salir.

nota: tomado del libro **Estructura** de datos, serie schaum McGraw-Hill,
pagina: 334 - 335, capítulo: 8 Grafos y sus aplicaciones, autor: Seymour
Lipschutz

Arboles de recubrimiento mínimo (búsqueda del camino más corto):

CAMINOS MINIMOS EN GRAFOS

Para lograr el propósito del recorrido mínimo dentro de un grafo **G**, es necesario para nuestro caso en particular (puesto que no es la única técnica existente) la utilización del algoritmo de **WARSHALL** para el camino mínimo, el cual se expresa de la forma siguiente:

Sea **G** un grafo con **m** nodos, **u 1** , **u 2** , ..., **u m** suponga que queremos encontrar la matriz de caminos **P** para el grafo **G**. **WARSHALL** dio un algoritmo

para este propósito que es mucho más eficiente que calcular las potencias de la matriz de adyacencia A y aplicar la proposición:

$$B_m = A + A^2 + A^3 + \dots + A^m$$

donde sea A la matriz de adyacencia y $P = P_{ij}$ la matriz de caminos de un grafo G entonces, $P_{ij} = 1$ si y solo si hay un número positivo en la entrada ij de la matriz

Este algoritmo de WARSHALL se usa para calcular el camino mínimo y existe un algoritmo similar para calcular el camino mínimo de G cuando G tiene peso.

Algoritmo de WARSHALL:

Un grafo dirigido G con M nodos está en memoria por su matriz adyacente A , este algoritmo encuentra la matriz de caminos (Booleana) P del grafo G .

1. [Inicializar P] repetir para $I, J = 1, 2, \dots, M$:

 si $A[I, J] = 0$, entonces: hacer $P[I, J] := 0$;

 si no: hacer $P[I, J] := 1$.

 [fin de bucle]

2. [Actualizar P] repetir paso 3 y 4 para $K=1, 2, \dots, M$:

3. repetir paso 4 para $I=1, 2, \dots, M$:

4. repetir para $J=1, 2, \dots, M$:

 hacer $P[I, J] := P[I, J] \vee (P[I, J] \wedge P[K, J])$.

 [fin de bucle]

 [fin de bucle paso 3]

 [fin de bucle paso 2]

5. Salir.

nota: tomado del libro Estructura de datos, serie schaum Mcgraw-Hill,

pagina: 322, capítulo: 8 Grafos y sus aplicaciones, autor: Seymour Lipschutz

Algoritmo de matriz de camino mínimo:

Cuando se trata de un grafo con peso G de M nodos está memoria mediante su matriz de peso W ; este algoritmo encuentra la matriz Q tal que $[I, J]$ es la longitud del camino mínimo del nodo VI al nodo VJ . INFINITO es un número muy grande y MIN es la función del valor mínimo.

1. [Inicializar Q] repetir para I, J=1, 2, . . . , M:

si $W [I, J] = 0$, entonces: hacer $Q [I, J] := \text{INFINITO}$;

si no: hacer $Q [I, J] := W [I, J]$.

[fin de bucle]

2. [Actualizar Q] repetir pasos 3 y 4 para K=1, 2, . . . , M:

3. repetir paso 4 para I = 1, 2, . . . , M:

4. repetir para J = 1, 2, . . . , M:

hacer $Q [i, J] := \text{MIN}(Q [i, J] + Q [i, K] + Q [K, J])$.

[fin de bucle]

[fin de bucle del paso 3]

[fin de bucle del paso 2]

5. Salir.

nota: tomado del libro *Estructura de datos, serie schaum* McGraw-Hill,

pagina: 324, capitulo: 8 Grafos y sus aplicaciones, autor: Seymour Lipschutz

Enunciado para ejemplo:

Dado un grafo simple no dirigido, conexo y ponderado de n nodos etiquetados con los números naturales desde el 1 hasta el n , se numeran los ejes desde 1 hasta m de acuerdo con el orden. Dados a continuación dos nodos cualesquiera, se trata de encontrar el camino más corto entre ambos nodos, utilizando el algoritmo de Dijkstra.

Entrada: En la primera línea, un número natural que indica el número de casos que se van a plantear. Para cada caso, una línea con el número de nodos n del

grafo, y la representación decimal del mismo (entero menor que $2^{\binom{n}{2}}$) separados por un blanco. En la siguiente línea, separados por blancos, m números naturales que representan los pesos de los ejes del grafo. En la siguiente línea, otro número natural p nos dice cuantos pares de nodos se van a proponer, y a continuación aparecen en líneas diferentes y separados por blancos todas estas parejas.

Salida: Para cada uno de los casos propuestos, el fichero de salida contendrá una línea indicando el caso de que se trata en la forma Grafo # con el símbolo # sustituido por el número del caso. Las siguientes m líneas contendrán la lista de adyacencias del grafo en la forma:

No.delnodo Nodoadyacente pesodeleje Nodoadyacente Pesodeleje...

siempre separando con blancos y con los nodos adyacentes en orden creciente de número. A continuación, **p** líneas que resuelven las **p** parejas de nodos planteadas, componiendo cada línea en la forma:

Pesodelcamino... ...nodosintermedios... ...nodofinal...

Ejemplo de Entrada:

2

4 49

53 82 53

2

1 2

1 3

8 14728196

81 48 30 64 71 13 91 10 65

3

2 1

4 1

8 1

Ejemplo de Salida:

Grafo 1

1 2 53

2 1 53 4 82

3 4 53

4 2 82 3 53

53 1 2

188 1 2 4 3

Grafo 2

1 4 81

2 6 48 7 30 8 64

3 4 71 6 13

4 1 81 3 71 8 91

5 6 10 7 65

6 2 48 3 13 5 10

7 2 30 5 65

8 2 64 4 91

213 2 6 3 4 1

81 4 1

172 8 4 1

Algoritmo de Dijkstra: Este algoritmo construye el árbol de caminos de longitud mínima entre un vértice fijado **V** y los restantes vértices en un grafo ponderado.

Observaciones:

1. Los pesos de las aristas deben ser no negativos.
2. El algoritmo de Dijkstra NO proporciona un árbol generador mínimo.

Ordenación Topológica:

Hasta recientemente todos los trabajos sobre Topología (Digital principalmente) se basaban en un enfoque de **Teoría** de Grafos. Este enfoque presenta, sin embargo, el problema de determinar la relación de adyacencia más razonable en **Zn**. Actualmente existen enfoques alternativos basados en nociones de Topología General. En este caso haremos una introducción a algunos de estos enfoques.

Topología definición:

1) Rama de la **matemática** que estudia las propiedades del espacio que son invariantes por homeomorfismos. Se trata de propiedades no métricas, es decir, de propiedades cualitativas, y no cuantitativas, lo que la distingue de la **geometría** común. Se la suele denominar "**geometría débil**" o "**geometría del caucho**". Por ejemplo, una circunferencia es topológicamente equivalente a un cuadrado, por más que sus propiedades métricas sean diferentes

2) Una topología en un conjunto X es una familia de subconjuntos de X que satisface ciertos axiomas (ver espacio topológico).

En esta sección estudiaremos las diferentes estrategias que se han planteado principalmente motivadas por problemas en el área del reconocimiento de formas para dotar a la digitalización D de un conjunto, de una estructura, no necesariamente explícitamente topológica, en términos de la cual formular propiedades de D relacionadas con las propiedades de la imagen original.

Las imágenes 2-dimensionales son las más ampliamente estudiadas, aunque últimamente las 3-dimensionales están siendo muy utilizadas. También se utilizan imágenes 4-dimensionales para representar imágenes 3-dimensionales en movimiento.

De las estrategias planteadas, la primera y la más utilizada es la introducida por Azriel Rosenfeld en 1970. Se basa en la noción de adyacencia entre puntos de Z^n (además de Z^n también considera en ocasiones los centros de una teselación del plano por hexágonos). Su enfoque descansa principalmente en nociones de Teoría de Grafos.

Desde entonces la Topología Digital ha proporcionado los fundamentos teóricos para importantes operaciones de procesamiento de imagen, como recuento de objetos, adelgazamiento de imágenes, detección de bordes y relleno de contornos. El adelgazamiento de imágenes es una operación de preprocesamiento en reconocimiento de formas. Su objetivo es reducir el conjunto de puntos de la imagen sin alterar la topología de la misma.

Ordenación topológica:

Suponga que S es un grafo tal que cada nodo v_i de S representa una tarea y cada arista (u, v) significa que la finalización de la tarea u es un pre-requisito para que comience la tarea v . Suponga que tal grafo S contiene un ciclo tal que:

$$P=(u, v, w, u)$$

Esto significa que no podemos empezar v hasta completar u , no podemos empezar w hasta terminar v y no podemos empezar u hasta completar w . Así no podemos completar ninguna tarea del ciclo. Por tanto, un grafo S así, que representa tareas y relaciones de precedencia, no puede tener ciclos.

Ahora suponga que S es un grafo sin ciclos, considere la relación $<$ sobre S definida como sigue:

$$u < v \text{ si existe un camino de } u \text{ a } v$$

Esta relación tiene las siguientes tres propiedades:

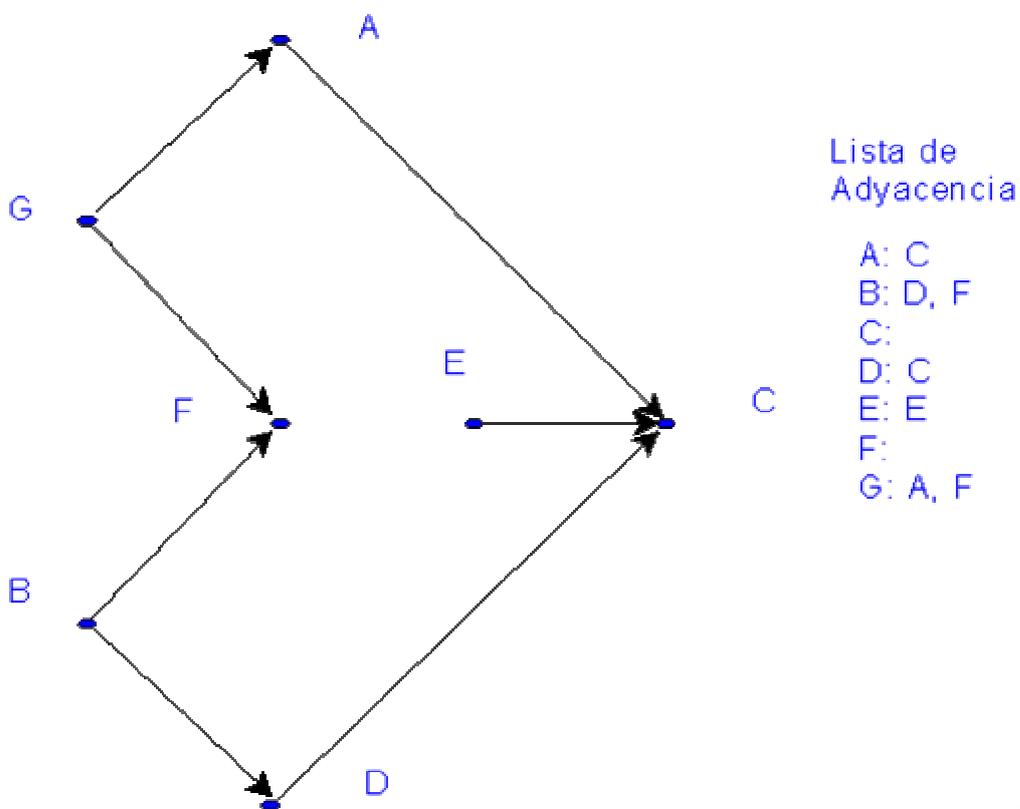
1. Para cada elemento u de S , tenemos que $u < u$. (Irreflexividad)
2. Si $u < v$, entonces $v < u$. (Asimetría)
3. Si $u < v$ y $v < w$, entonces $u < w$. (Transitividad)

Tal relación $<$ sobre S se llama ordenación parcial de S , y S con tal ordenación se llama conjunto parcialmente ordenado o conjunto **po**. Así, un grafo S sin ciclos se puede considerar un conjunto parcialmente ordenado.

Por lo tanto, puede también suponer que si S es un conjunto parcialmente ordenado con la ordenación parcial denotada por $<$, entonces S se puede considerar un grafo cuyos nodos son los elementos de S y cuyas aristas están definidas como sigue:

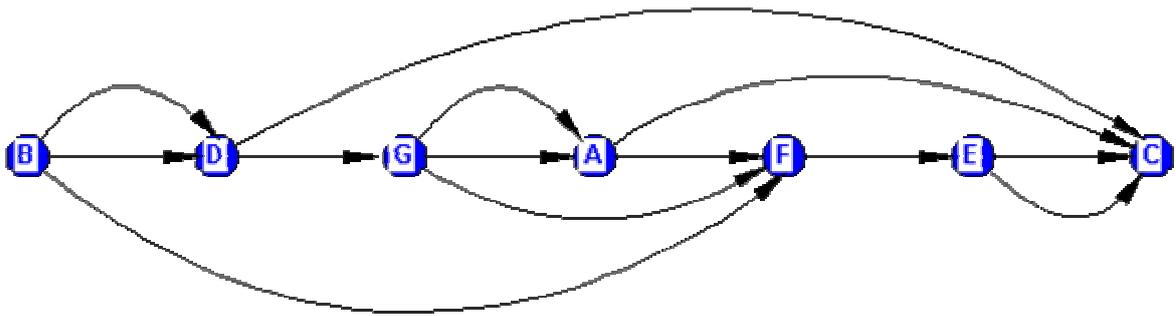
(u, v) es una arista en S si $u < v$

más aun, se puede demostrar que un conjunto S parcialmente ordenado, considerado como un grafo, no tiene ciclos.

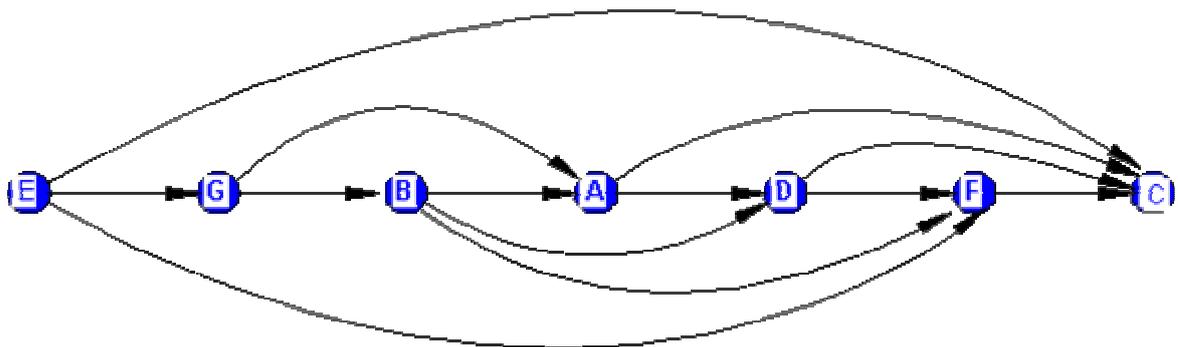


Como ejemplo podemos plantear que: sea S el grafo de la figura, observe que S no tiene ciclos. Así S se puede considerar un conjunto parcialmente ordenado. Note que $G < C$, ya que existe un camino desde G hasta C . Similarmente, $B < F$ y $B < C$. Por otro lado B no es menor que A , ya que no existe camino de B a A , al igual que A no es menor que B .

Por lo tanto; sea S un grafo dirigido sin ciclos (o conjunto parcialmente ordenado). Una ordenación topológica T de S es una ordenación lineal de los nodos de S que preserva la ordenación parcial de S original, o sea, que si $u < v$ en S (si hay un camino de u a v en S), entonces u va delante de la v en la ordenación lineal T , este se muestra en la siguiente figura, donde se incluyen las aristas para indicar que concuerdan con la dirección de la ordenación lineal.



Dos modelos de ordenación topológica.



Información adicional sobre Topología:

Topología combinatoria:

Rama de la topología que reduce el estudio de curvas y superficies a ciertos esquemas determinados por polígonos curvilíneos, evitando de esta forma pensarlas como **conjuntos** de puntos, como lo hace la topología conjuntista. El tratamiento combinatorio es más cercano al **álgebra**, y reduce el **concepto** de homeomorfismo a unas pocas reglas que permiten decidir cuándo dos esquemas combinatorios son equivalentes.

Topología inducida:

Dado un subconjunto A de un espacio topológico X, se llama topología inducida a la topología definida en A que toma como abiertos a todos los **conjuntos** de la forma $U \cap A$, en donde U es un abierto de X. En estas condiciones, se dice que A es un subespacio de X.

Topología usual:

La topología usual del espacio n–dimensional (R^n) tiene como abiertos básicos a las bolas n–dimensionales (abiertas). Es decir, un conjunto de R^n es abierto si y sólo si es unión de cierto número de bolas abiertas. Equivalentemente, diremos que A es abierto si y sólo si para todo punto $x \in A$ existe una bola B contenida en A tal que $x \in B$ (A es entorno de x).

Toro:

Se llama así a la superficie de **revolución** engendrada por la rotación de una circunferencia en **torno** a un eje que no la toque en ninguno de sus puntos. Si bien esta definición es geométrica, las propiedades topológicas del toro son de gran importancia. En especial, la **propiedad** de tener un asa, o agujero, que determina que existan en el toro lazos no reducibles. Un importante teorema de la topología combinatoria asegura que toda superficie cerrada y orientable es un toro con n agujeros. El caso $n = 0$ corresponde obviamente a la esfera, si se la piensa como un "toro sin agujeros", y el caso $n = 1$ es el toro usual. Si bien la definición habitual del toro lo presenta como una superficie sumergida en el espacio tridimensional, es fácil ver que es homeomorfo al **producto** cartesiano de dos circunferencias, sumergido en R^4 (espacio cuatridimensional). Es decir, la definición topológica del toro es: $T_2 = S^1 \times S^1$. Esto permite generalizar, y definir al toro n -dimensional como el **producto** cartesiano de n circunferencias, es decir: $T_n = S^1 \times \dots \times S^1$.

En la topología combinatoria, el toro bidimensional se define identificando dos a dos los lados opuestos de un rectángulo, como **muestra** la figura:

Algoritmo de ordenación topológica:

El siguiente algoritmo encuentra una ordenación topológica **T** de un grafo **S** sin ciclos.

1. Encontrar el grado de entrada $GraEnt(N)$ de cada nodo N de S (se puede hacer recorriendo cada lista de adyacencia)
2. Poner en una cola todos los nodos con grado de entrada nulo.
3. Repetir los pasos 4 y 5 hasta que la cola esté vacía.
4. Quitar el nodo N al frente de la cola (haciendo $Frente := Frente + 1$)
5. Repetir lo siguiente para cada vecino M de N :

Hacer $GraRnt(M) := GraEnt(M) - 1$ (esto elimina la arista de N a M)

si $GraEnt(M) = 0$, entonces: Añadir M al final de la cola.

[fin de bucle paso 5]

[fin de bucle paso 3]

6. Salir.

nota: tomado del **libro** Estructura de datos, serie schaum Mcgraw-Hill,

pagina: 340, capitulo: 8 Grafos y sus aplicaciones, autor: Seymour Lipschutz

A N E X O

TEMAS AFINES Y COMPLEMENTARIOS

Caminos y Conexión: Un camino en un grafo es una sucesión de vértices y aristas de la forma $v_0 a_1 v_1 a_2 \dots v_{k-1} a_k v_k$ donde la arista a_i une los vértices v_{i-1} y v_i . Éste es un camino de v_0 a v_k , de longitud k , siendo v_1, \dots, v_{k-1} los vértices interiores del camino. Si $v_0 = v_k$ decimos que el camino es cerrado. Un ciclo es un camino cerrado con todas sus aristas distintas y con todos sus vértices distintos (salvo, claro es, los extremos $v_0 = v_k$).

Propiedades:

1. El nº de caminos de longitud k de v_i a v_j es el elemento ij de la matriz $M(G)^k$.
2. Un grafo G es bipartido $\iff G$ no tiene ciclos de longitud impar.
3. Se llama distancia entre dos vértices u y v , a la longitud mínima de un camino que conecta dichos vértices y se designa por $d(u,v)$.
4. Se llama diámetro de G a la máxima distancia entre dos vértices de G , $\text{diam}(G)$.

Un grafo es conexo si para cada par de vértices u y v existe un camino de u a v . Si G es un grafo no conexo (o desconexo), cada uno de sus subgrafos conexos maximales se llama componente conexa de G .

Un vértice v se llama vértice-corte (o punto de articulación) de G si el grafo $G - \{v\}$ tiene más componentes conexas que G .

Una arista a de un grafo G se llama puente si $G - \{a\}$ tiene más componentes conexas que G .

Conexo: un espacio topológico X se dice conexo si no contiene ningún subconjunto abierto y cerrado, excepto \emptyset y X . Intuitivamente, un conjunto es conexo cuando no está compuesto por dos o más partes separadas. Una definición mucho más fácil de entender es la de conjunto arcoconexo. Sin embargo, se puede probar que ambas nociones no coinciden: todo conjunto arcoconexo es conexo, pero la recíproca es falsa. En la topología usual, todo abierto conexo es también arcoconexo.

Espacio n–dimensional: se llama espacio n–dimensional usual al conjunto \mathbb{R}^n , construido como el producto cartesiano $\mathbb{R} \times \dots \times \mathbb{R}$ (n veces), en donde \mathbb{R} es

el conjunto de los números reales. Los elementos de \mathbf{R}^n se piensan como **vectores** de n coordenadas. El vector nulo es aquel cuyas coordenadas son todas 0, y se lo llama "origen" o "centro de coordenadas". Por ejemplo, el plano \mathbf{R}^2 es el conjunto de todos los pares ordenados (x, y) en donde sus dos coordenadas x, y son números reales cualesquiera, y su origen es el vector $(0, 0)$. A este espacio se le suele asignar una topología, conocida como topología usual de \mathbf{R}^n .

Espacio topológico: se llama espacio topológico a un conjunto X provisto de una topología, es decir, una **familia** de subconjuntos de X , llamados abiertos, que satisfacen los siguientes axiomas:

1. \mathcal{A} y X son **conjuntos** abiertos
2. La intersección de un número finito de **conjuntos** abiertos es un conjunto abierto
3. La unión de cualquier número de conjuntos abiertos es un conjunto abierto

Se desprende de la definición que en cualquier espacio topológico X los conjuntos \mathcal{A} y X son a la vez abiertos y cerrados (ver también: topología usual)

Función continua: dados dos espacios topológicos X e Y , la función $f: X \rightarrow Y$ se dice continua en un punto $a \in X$ si dado un entorno V del punto $f(a) \in Y$, existe un entorno U de a tal que $f(U) \subseteq V$, es decir, $f(x) \in V$ para todo $x \in U$. Esto puede expresarse mediante la noción de límite: f es continua en a si

$$\lim_{x \rightarrow a} f(x) = f(a)$$

En la topología usual, la noción de continuidad en a equivale a la **propiedad** de que si $\{x_n\}$ es cualquier sucesión en X que converge a a , entonces la sucesión $\{f(x_n)\}$ converge a $f(a)$. Intuitivamente, podemos decir: "cuanto más se acerca x_n a a , más se acerca $f(x_n)$ a $f(a)$ ". f se dice continua cuando es continua en todos sus puntos. Equivalentemente, f es continua si y sólo si la **imagen** inversa de un abierto cualquiera es un conjunto abierto.

Grupo fundamental: dado un espacio topológico X , se puede formar el conjunto de todos los lazos en X que salen de un cierto punto, considerando como equivalentes a dos lazos que se puedan superponer mediante una homotopía (es decir, tales que se pueda deformar a uno de ellos en forma continua hasta obtener el otro). A dicho conjunto se le asigna una estructura (algebraica) de **grupo**, lo que determina el llamado **grupo** fundamental de X . Se trata de un invariante topológico muy útil. Por ejemplo, el **grupo** fundamental de una circunferencia es \mathbf{Z} , el conjunto de los números enteros ($\mathbf{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$), hecho que resulta claro pues todo lazo cerrado sobre la circunferencia está determinado unívocamente por la cantidad de vueltas, y el sentido en que se las recorre. El **grupo** fundamental de un toro es $\mathbf{Z} \times \mathbf{Z}$, pues en este caso deben tenerse en cuenta las vueltas dadas "alrededor" del agujero, y también "a través" del mismo. Este resultado es, claro está, coherente con el hecho de que el toro puede pensarse como el **producto** cartesiano de dos circunferencias (ver: toro).

Homeomorfismo: se llama homeomorfismo entre dos espacios topológicos X e Y a una función $f: X \rightarrow Y$ que resulte biunívoca y bicontinua, es decir:

f es "uno a uno" (biunívoca), lo que significa que para cada elemento $x \in X$ existe un único $y \in Y$ tal que $f(x) = y$ y viceversa. Esto permite definir la función inversa, $f^{-1}: Y \rightarrow X$

f y f^{-1} son continuas (f es bicontinua)

La noción de homeomorfismo responde a la idea intuitiva de "deformación", y determina cierta clase de equivalencia: dos espacios homeomorfos tienen las mismas propiedades topológicas.

Homología: invariante topológico que asocia a cada espacio topológico una estructura algebraica llamada "complejo". Como invariante, tiene mayor precisión que el grupo fundamental, aunque su definición y cálculo resultan más complicados.

Homotopía: dados dos espacios topológicos X e Y , una homotopía es una función continua $h: X \times [a, b] \rightarrow Y$, en donde $[a, b]$ es un intervalo cerrado. Por comodidad, siempre supondremos que $[a, b]$ es el intervalo $[0, 1]$. Se puede interpretar intuitivamente la noción de homotopía pensando al $[0, 1]$ como un intervalo de tiempo, y en consecuencia h representa una cierta deformación a partir del instante inicial $t = 0$, hasta llegar a $t = 1$ pasando por cada instante t fijo.

Identificar: operación topológica que responde a la noción intuitiva de "pegar". Consiste en definir alguna relación de equivalencia entre puntos de un espacio topológico X , lo que permite definir el espacio cociente. Por ejemplo, si se identifican uno a uno los puntos de dos lados opuestos de un rectángulo, se obtiene una superficie tubular similar a un "cinturón", o una porción de cilindro. En cambio, si esta identificación se efectúa orientando a los dos lados en sentidos opuestos, se obtiene una Banda de Möbius.

Interior: dado un conjunto A , se llama interior de A al mayor abierto contenido en A . Notación: $A^\circ = \text{interior de } A$. Por definición, es claro que un conjunto es abierto si y sólo si coincide con su interior. El interior de A se puede pensar como el conjunto de puntos de A que no pertenecen a su frontera, es decir: $A^\circ = A - \text{Fr}(A)$.

Intervalo: dados dos números reales $a < b$, se llama intervalo entre a y b al conjunto de puntos de la recta contenidos entre a y b . Caben cuatro posibilidades, según se incluya o no a cada uno de los extremos:

1. $(a, b) = \{ x \in \mathbb{R} / a < x < b \}$ (intervalo abierto)
2. $[a, b) = \{ x \in \mathbb{R} / a \leq x < b \}$ (intervalo semiabierto)
3. $(a, b] = \{ x \in \mathbb{R} / a < x \leq b \}$ (intervalo semiabierto)
4. $[a, b] = \{ x \in \mathbb{R} / a \leq x \leq b \}$ (intervalo cerrado)

También se definen los siguientes intervalos no acotados: $(a, +\infty)$, $[a, +\infty)$,

$(-\infty, b), (-\infty, b]$. Por ejemplo, $(a, +\infty) = \{x \in \mathbf{R} / x > a\}$. Los símbolos $+\infty$ y

$-\infty$ responden únicamente a una mayor simplicidad en la **escritura**, ya que no se trata de números reales. Por esa razón, todo intervalo no acotado es abierto en su "extremo infinito". Obviamente, el intervalo $(-\infty, +\infty)$ equivale a toda la recta \mathbf{R} . Es fácil ver que cualquier intervalo abierto es homeomorfo a \mathbf{R} .

Invariante: se llama invariante topológico a aquellas propiedades de un espacio topológico que permanecen cuando se le aplica un homeomorfismo. Algunos invariantes muy conocidos son la compacidad, la conexión, el grupo fundamental, la homología, etc. En general, cada **teoría matemática** tiene sus propios invariantes: así, los invariantes geométricos son las propiedades que conserva una figura cuando se le aplica una rotación o una traslación (movimientos rígidos).

Matrices: la matriz de adyacencia de un grafo G con n vértices $\{v_1, \dots, v_n\}$ es la matriz $n \times n$, $M(G) = (a_{ij})$, donde a_{ij} es el nº de aristas que unen v_i con v_j . La matriz de incidencia de un grafo simple G con n vértices $\{v_1, \dots, v_n\}$ y k aristas $\{e_1, \dots, e_k\}$ es la matriz $n \times k$, $I(G) = (b_{ij})$, donde $b_{ij} = 1$ si v_i es incidente con e_j y $b_{ij} = 0$ en caso contrario.

Plano proyectivo: espacio definido en **geometría** proyectiva, de acuerdo con la idea intuitiva de agregar al plano euclidiano un "horizonte", de modo tal que dos rectas paralelas determinen un (único) punto. Las rectas resultan entonces cerradas, es decir, homeomorfas a una circunferencia, hecho relacionado además con la **propiedad** que tiene el plano proyectivo de ser compacto. Al horizonte, que también es una recta, se lo suele llamar "recta impropia", pues está compuesta de puntos impropios, también llamados puntos "del infinito".

En la **geometría** proyectiva los conceptos de "punto" y "recta" son duales, puesto que pueden intercambiarse. Por ejemplo, el enunciado: "Dos puntos determinan una única recta" se transforma en su dual "Dos rectas determinan un único punto", que también es válido, aunque no lo es en la geometría euclidiana.

Poliedro topológico: generalización de la noción geométrica de poliedro. Consiste en un **sistema** formado por un número finito de polígonos topológicos sujetos a ciertas condiciones, entre las cuales se tiene, por ejemplo, que dos polígonos distintos no tienen puntos interiores comunes, que los lados de los polígonos del **sistema** coinciden dos a dos, etc.

Polígono topológico: generalización de la noción geométrica de polígono. Consiste en tomar cierto número finito $n > 1$ de puntos en una circunferencia. Los arcos así determinados serán los lados, y los puntos se llamarán vértices del polígono. El polígono estará formado entonces, por el conjunto de lados y la región interior a la circunferencia.

Recorridos en un Grafo: Un camino euleriano en un grafo es un camino que contiene a todas las aristas del grafo exactamente una vez. Un grafo es euleriano si contiene un camino euleriano cerrado.

Teorema: Un grafo conexo G es euleriano \hat{U} Todos los vértices de G tienen grado par.

Consecuencia: Un grafo conexo G tiene un camino euleriano no cerrado \hat{U} G tiene, exactamente, dos vértices de grado impar.

Algoritmo de Fleury (para construir un camino euleriano cerrado en un grafo euleriano).

Paso 1.- Se comienza en un vértice cualquiera v_0 .

Paso 2.- Si se ha construido el camino $v_0 a_1 v_1 a_2 \dots v_{k-1} a_k v_k$ con aristas distintas, se elige la arista siguiente a_{k+1} con las condiciones: (1) a_{k+1} incidente con v_k y (2) no ser puente en el grafo $G - \{a_1, a_2, \dots, a_k\}$ (salvo que no haya alternativa).

Paso 3.- Se sigue hasta que el camino contenga todas las aristas.

Un camino hamiltoniano en un grafo es un camino que contiene a todos los vértices del grafo exactamente una vez (salvo $v_0 = v_n$, si el camino es cerrado). Un grafo hamiltoniano es aquel que contiene un ciclo hamiltoniano.

Propiedad: Un grafo bipartido $G = (V_1 \dot{\cup} V_2, A)$ con $|V_1| \neq |V_2|$ no es hamiltoniano.

Teorema: Sea G un grafo simple de n vértices. Si para todo par de vértices x e y no adyacentes se cumple que $d(x) + d(y) \geq n$, entonces G es hamiltoniano.

Teorema: Si G es un grafo hamiltoniano entonces, para todo $S \subset V$ se cumple que el número de componentes conexas de $G - S$, es menor o igual que $|S|$.

Observación: NO hay caracterización para los grafos hamiltonianos.

Bibliografía:

1. Balabanian, N.: **Circuitos Eléctricos**. MacGraw Hill. 1994. 127 – 135.
2. Balabanian, N.; Bickart, T.A.; Seshu, S.: Teoría de **Redes Eléctricas**. Reverté, 1972. 200 – 204.
3. Budak, A. Passive and Active Network Analysis and Synthesis. Houghton Mifflin, 1974. 97 – 140.
4. Lipschultz, Seymour: Estructura de Datos teoría y **problemas**. Schaum-McGraw-Hill. 1988. 315 – 357.
5. Folk, M y Zoellick, B.: Estructura de **Archivos**, Addison-Wesley, Reading, MA, 1992. 420 – 423.
6. Cormen, Leiserson, Rivest: Introducción a la Algoritmica, The MIT Press-Mc Graw Hill, 1990. 199 – 215.

7. A.Giraldo, Topología Digital, Prepublicaciones del Departamento de **Matemática** Aplicada, FIM/2/DMA/97, Facultad de **Informática**, UPM, 1997.
8. A.Giraldo, Digitizations preserving shape, Vision Geometry VI, Proc. of the 1997 SPIE Conference on Vision Geometry, San Diego, 1997.
9. G.T.Herman, **Análisis de la Imagen** en Aplicación topológica, **Visión de la Computadora**, Procesando Gráficos e Imagen, 52, 1990, 409-415.
10. E.Khalimsky, Topological structures in computer science, J. Appl. Math. Simulation, 1, 1987, 25-40.
11. T.Y.Kong, R.Kopperman y P.R.Meyer, A Topological Approach to Digital Topology, American Mathematical Monthly, 98, 1991, 901-917.1.
12. T.Y.Kong, R.Kopperman y P.R.Meyer (eds.), Problema especial en Topología Digital, Topología y sus Aplicaciones, 46, 1992.
13. T.Y.Kong y A.Rosenfeld, Digital Topology: Introduction and Survey, Computer Vision, Graphics and Image Processing, 48, 1989, 357-393.
14. T.Y.Kong y A.Rosenfeld (eds.), Problema especial en Topología y Geometría en Visión de la **Computadora**, **Periódico de Imagen Matemático y Visión**, 6, 1996.
15. V.A.Kovalevsky, Finite Topology as Applied to Image Analysis, Computer Vision, Graphics and Image Processing, 46, 1989, 141-161.
16. E.H.Kronhemeir, Alternativas topológicas Digitales, Topología y sus Aplicaciones, 46, 1992, 269-277.
17. Knuth D.E.; Clasificación y búsqueda. El **Arte** de Programar Ordenadores Vol. III. Ed. Revert S.A., 1987.
18. Knuth D.E.; **Algoritmos** Fundamentales. El **Arte** de Programar Ordenadores Vol. I. Ed. Revert S.A., 1980.
19. Aho A. V., Hopcroft J.E., Ullman J.D.; **Estructuras de Datos y Algoritmos**. Ed. Addison-Wesley, 1988.
20. Deitel H.M., Deitel P.J.; C++ How To Program. Ed. Prentice Hall, 1994.

Trabajo realizado por:

Felipe Costales

felcos@cantv.net

Fuente: <http://www.monografias.com/trabajos/grafos/grafos.shtml>