

## GRAFOS.

### INTRODUCCIÓN

El “gráfico” tiene varios sentidos en matemáticas. Hemos usado el término “gráfica” en el sentido de una relación o de una función.

En muchas partes de la ciencia de las computadoras y de la informática aparecen los grafos, especialmente los grafos de árbol, y los grafos dirigidos. Los diagramas de flujo, por ejemplo, son grafos dirigidos.

### GRAFOS Y MULTIGRADOS.

Un **grafo** consta de dos cosas:

- ☞ Un conjunto  $N$  cuyos elementos se llaman **nod**os, **puntos** o **vértices**.
- ☞ Un conjunto  $S$  de parejas no ordenadas de nodos diferentes, llamadas **segmentos**, **aristas** o **arcos**.

Denotaremos un grafo por  $G=(N, S)$  cuando queremos destacar las dos partes de  $G$ .

Los nodos  $u$  y  $v$  se llaman **adyacentes** si hay un segmento  $\{u, v\}$

Representaremos de una manera natural los grafos por diagramas en el plano. O sea, cada nodo  $v$  de  $N$  se representa por un punto (o pequeño círculo) y cada segmento  $s = \{v_1, v_2\}$  se representa por una curva que conecta sus **terminales**  $v_1$  y  $v_2$ .

### EJEMPLO.

La figura 1.1 representa el  $G$  con cuatro vértices, **A**, **B**, **C** y **D**, y cinco **segmentos**  $s_1 = \{A, B\}$ ,  $s_2 = \{B, C\}$ ,  $s_3 = \{C, D\}$ ,  $s_4 = \{A, C\}$ ,  $s_5 = \{B, D\}$ . Usualmente denotamos un grafo dibujando su diagrama en lugar de hacer una lista explícita de sus nodos y segmentos.

La figura 1.2 no es un grafo sino un **multigrafo**. La razón es que  $s_4$  y  $s_5$  son segmentos múltiples, o sea segmentos que conectan las mismas terminales, y  $s_6$  es un **lazo**, o sea, un segmento cuyas terminales son el **mismo nodo**. La definición de grafo **no permite ni segmentos múltiples o multisegmentos, ni lazos**. En otras palabras, podemos definir un grafo como **un multigrafo sin multisegmentos ni lazos**.

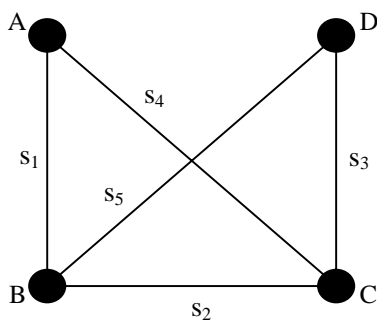


Figura 1.1

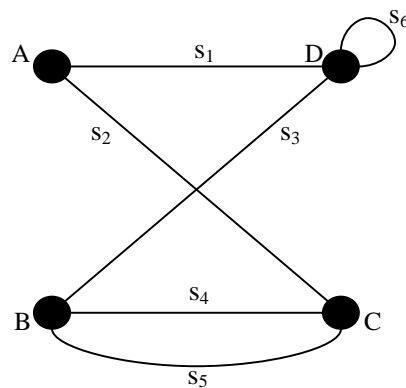


Figura 1.2

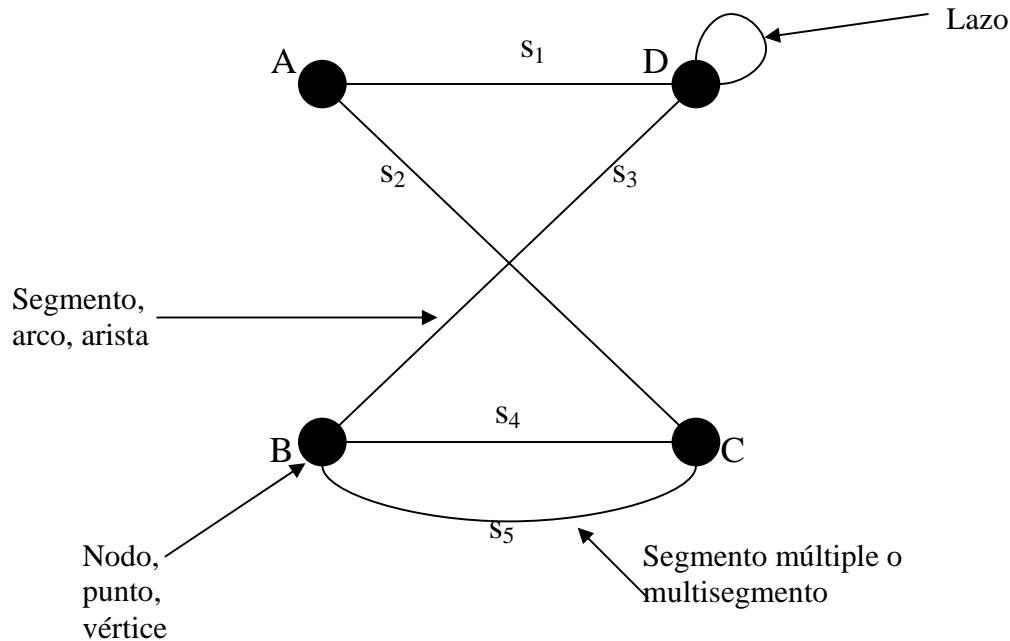


Figura 1.3

## GRADO DE UN NODO

Si  $v$  es una terminal de un segmento  $s$ , decimos que  $s$  es **incidente** en  $v$ . El **grado** de  $v$ , escrito  $\text{gr}(v)$ , es igual al número de segmentos que inciden en  $v$ . (Un **nodo de grado cero**, o sea un nodo que no pertenece a ningún segmento, se llama **nodo aislado**)

☞ **Teorema 1.-** La suma de los grados de los nodos de un grafo es igual al doble del número de segmentos.

## EJEMPLO

Observemos que en la figura 1.1 se tienen por cada vértice, los grados siguientes:

$$\text{gr}(A) = 2, \text{gr}(B) = 3, \text{gr}(C) = 3, \text{gr}(D) = 2$$

La suma de los grados es 10, que, como dice el Teorema 1, es el doble de segmentos. Se dice que un nodo es **par o impar** según que su grado sea par o impar. Así **A** y **D** son nodos pares, mientras que **B** y **C** son nodos impares.

Ahora veamos que en la figura 1.2 se tienen por cada vértice, los grados siguientes:

$$\text{gr}(A) = 2, \text{gr}(B) = 3, \text{gr}(C) = 3, \text{gr}(D) = 4$$

Como podremos notar el grado de **D** no es 2, sino 4, ya que el **lazo** se cuenta 2 veces para el grado de su nodo. Y la suma de los grados del nodo es 12, y una vez más se comprueba el Teorema 1, donde el grado total corresponde al doble de sus segmentos.

## VALENCIA

La valencia es la suma de los grados de los nodos.

De la figura 1.1

<b>A</b>	<b>2</b>
<b>B</b>	<b>3</b>
<b>C</b>	<b>3</b>
<b>D</b>	<b>2</b>
<b>Val</b>	<b>10</b>

De la figura 1.2

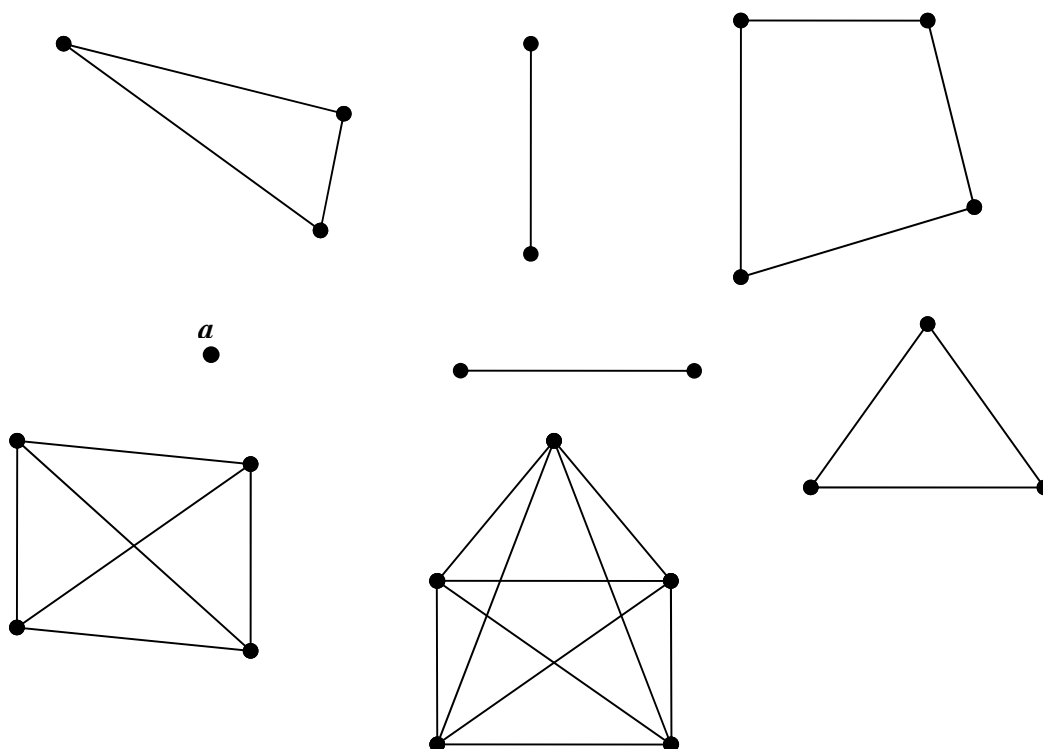
<b>A</b>	<b>2</b>
<b>B</b>	<b>3</b>
<b>C</b>	<b>3</b>
<b>D</b>	<b>4</b>
<b>Val</b>	<b>12</b>

## Tipo de grafos.

### Grafos simples:

Se dice que es *grafo simple* cuando no hay más de una arista entre un par de nodos (no más de una arista dirigida en el caso de gráficas dirigidas).

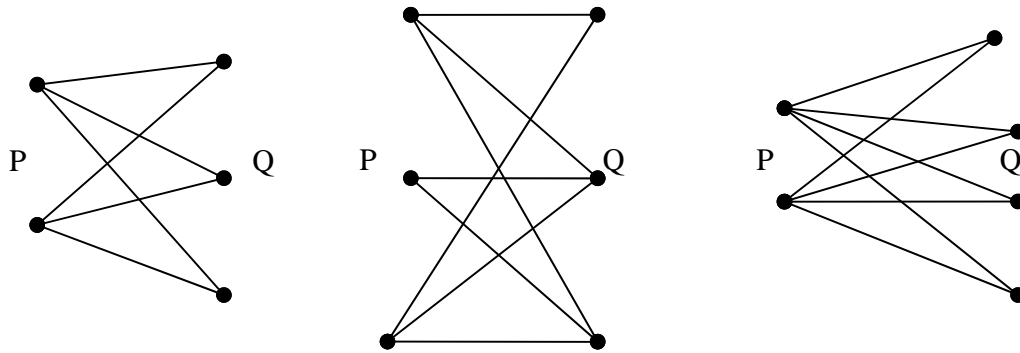
Ejemplos



### Grafos bipartitos:

Se dice que es un grafo bipartito  $G$  cuando un conjunto de nodos  $N$  se puede particionar en dos subconjuntos  $P$  y  $Q$  tales que, cada segmento de  $G$ , conecta un nodo de  $P$  con un nodo  $Q$ .

Ejemplos:



Un grafo *bipartito completo*, es cuando cada nodo de  $P$  está conectado con cada nodo de  $Q$ . En los ejemplos anteriores, todos son grafos *bipartitos completos*.

### Grafos Completos:

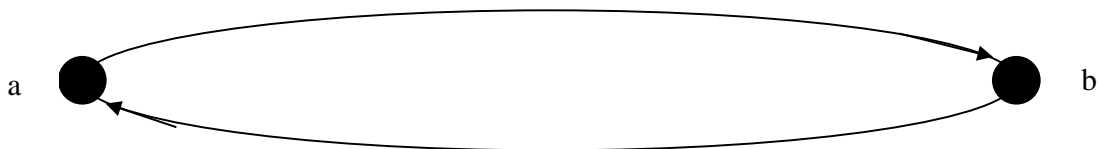
Un grafo completo es cuando cada nodo está conectado con otro nodo. Al grafo completo de  $n$  nodos se le denota  $K_n$ .

Ejemplos:

En el caso del grafo  $a$ , el cual es un nodo aislado, se dice que es un grafo trivial.

### Grafos de similitud:

Veamos la siguiente gráfica:



Aquí se muestran dos grafos  $(a, b)$  y  $(b, a)$  no son iguales pero son similares ya que en los dos grafos tienen los mismos vértices.

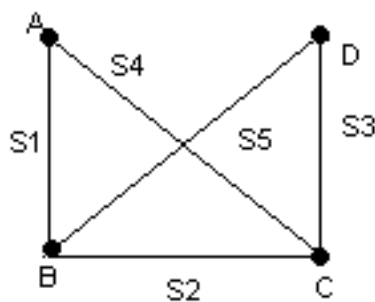
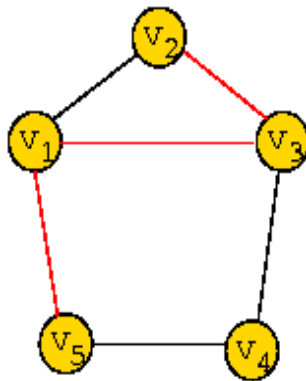
### Caminos.

Se llama caminos a una secuencia de aristas ( $V_1, V_2, V_3 \dots V_N$ ) de la manera que el vértice final de cada uno sirve de vértice inicial al siguiente.

**Camino** en un grafo es una sucesión de vértices y arcos.

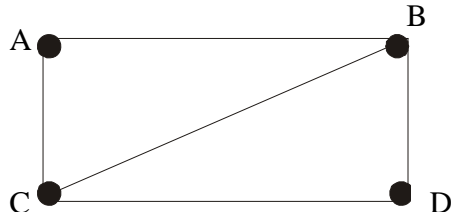
- Al número de arcos que atraviesa el camino se le denomina **Longitud del camino**.
- Camino de longitud 0 es aquel constituido únicamente por un vértice.
- **Extremos del camino** son los vértices inicial y final.
- **Vértices interiores** son aquellos situados entre los extremos del camino.

Un camino (en rojo)



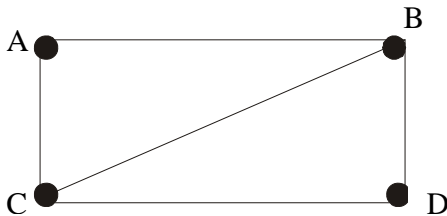
## Tipos de caminos.

**El camino elemental o trayectoria:** es un camino que pasa por una serie de vértices una sola vez. Es decir, es aquel que no pasa 2 veces por un mismo vértice, salvo, excepcionalmente, que el vértice que se repite sea el inicial y el final.



➤ Ejemplo: el camino (A, C, D, B, A)

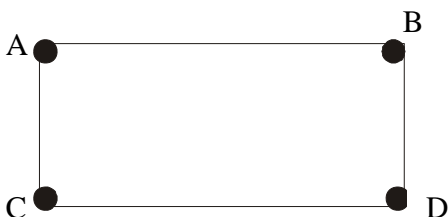
**Camino simple o sendero:** es un camino que pasa por una serie de aristas una sola vez. Todo camino elemental es un camino simple, pero la inversa puede no cumplirse.



➤ Por ejemplo: para ir de A a D, podría ser:  
(A, B, D), (A, C, D), (A, B, C, D) o (A, C, B, D)

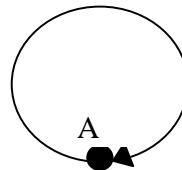
**Circuito o ciclos:** es un camino cerrado, el vértice final coincide con el vértice inicial. Un camino o un circuito se llaman *hamiltoniano* si pasa una sola vez por todos los vértices del grafo, y se denomina *euleriano* si pasa una sola vez por todas las aristas del grafo.

**Camino cerrado** es aquel cuyo vértice final coincide con el vértice inicial.

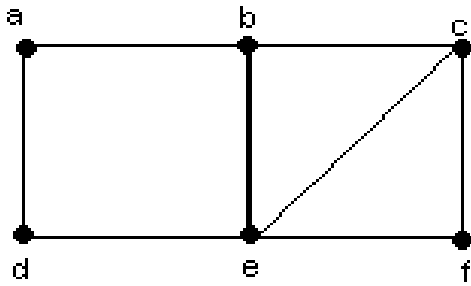


Por ejemplo: (A, C, D, B, A)

**Ciclo** es un camino simple, elemental y cerrado, de longitud positiva ( $n > 0$ ). El ciclo más elemental es un bucle (también reciben el nombre de *lazo* o *rizo*)



**Ejemplo:** Considerando el grafo mostrado a continuación. Encuentra:



- Todas las trayectorias del nodo *a* al nodo *f*.
- Todos los senderos de *a* a *f*.
- La distancia entre *a* y *f*,
- El diámetro del grafo.

- a) Una *trayectoria* de *a* a *f* es un camino tal que, ningún nodo y por lo tanto ningún segmento se repite. En este caso, hay siete de tales trayectorias.

(a, b, c, f)	(a, d, e, f)
(a, b, c, e, f)	(a, d, e, b, c, f)
(a, b, e, f)	(a, d, e, c, f)
(a, b, e, c, f)	

- b) Un *sendero* de *a* a *f* es un camino tal que no se repite ningún segmento. Observemos que hay nueve de tales senderos, las siete trayectorias descritas anteriormente en el punto a), y además se tienen:

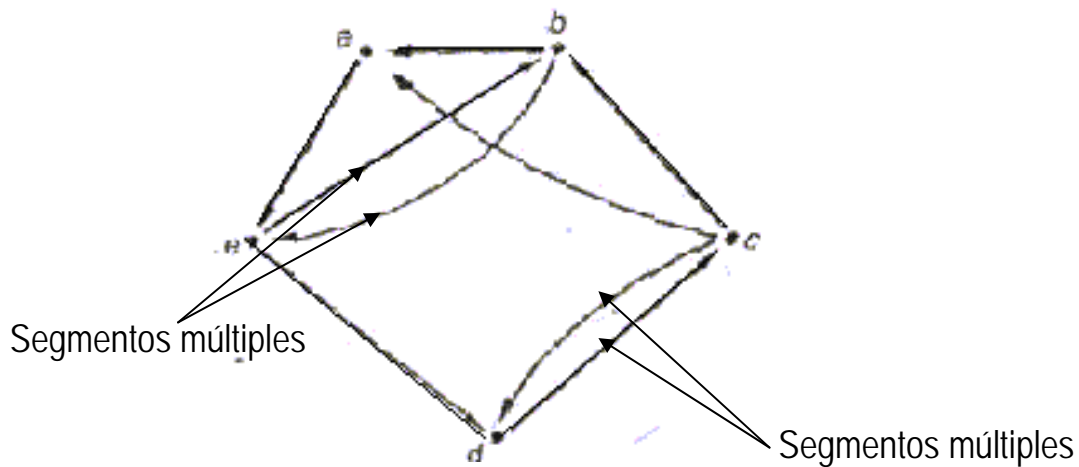
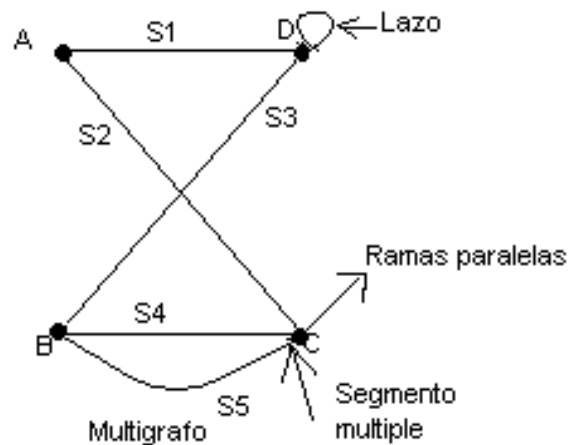
(a, d, e, b, c, e, f)    y    (a, d, e, c, b, e, f)

- c) La distancia de *a* a *f* es 3, ya que hay una trayectoria, (a, b, c, f) de *a* a *f* de longitud 3 y no hay ninguna trayectoria mas corta de *a* a *f*.
- d) La distancia entre dos nodos cualesquiera no es mayor que 5, y la distancia mayor entre *a* y *f* es 5; por lo tanto el diámetro del grafo es 5.

### Ramas paralelas.

Una *rama* es una trayectoria dirigida continua de un nodo a otro. También reciben el nombre indistintamente de *arcos*, *aristas* o *segmentos*.

Las ramas paralelas o segmentos múltiples, son aristas que conectan las mismas terminales. Es decir, que del mismo vértice parten 2 o más aristas a otro.

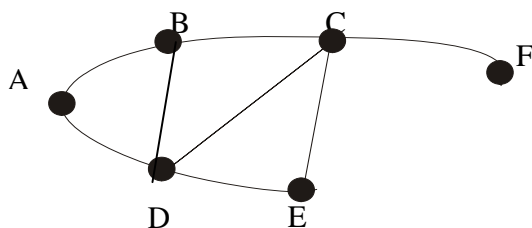


### Ejemplo:

Para el grafo de la siguiente figura, por ejemplo, tenemos los siguientes tres caminos abiertos. Podemos enumerar solamente las aristas o solamente los vértices (si el otro queda determinado claramente).

1.  $\{a, b\}, \{b, d\}, \{d, c\}, \{c, e\}, \{e, d\}, \{d, b\}$ : este es un camino de a-b de longitud 6 en el que se repite los vértices d y b, así como la arista  $\{b, d\} = \{d, b\}$ .
2.  $b \rightarrow c \rightarrow d \rightarrow e \rightarrow c \rightarrow f$ : aquí tenemos un camino b-f de longitud 5 donde se repite el vértice c, sin que permanezcan las aristas más de una vez.
3.  $\{f, c\}, \{c, e\}, \{e, d\}, \{d, a\}$ : en este caso el camino f-a tiene una longitud de 4, sin repetición de vértices o aristas.

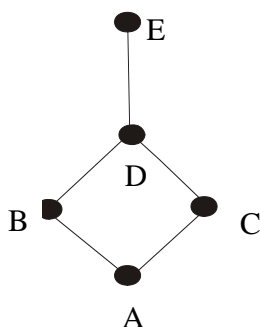




### Trayectoria y circuito de Euler

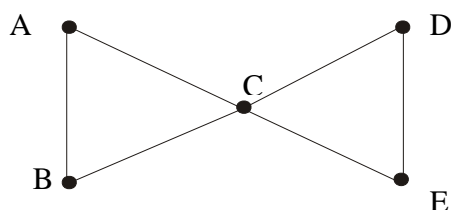
#### Circuito de Euler:

- a) Si una gráfica  $G$  tiene un vértice de grado impar, entonces no puede existir un circuito de Euler en  $G$ .



vértice	grado
A	2
B	2
C	2
D	3
E	1

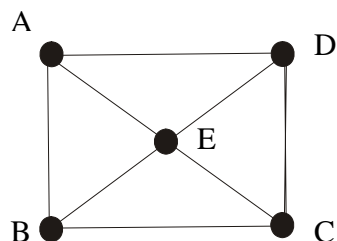
- b) Si  $G$  es una gráfica conexa y todos los vértices tienen grado par, entonces existe un circuito de Euler.



vértice	grado
A	2
B	2
C	4
D	2
E	2

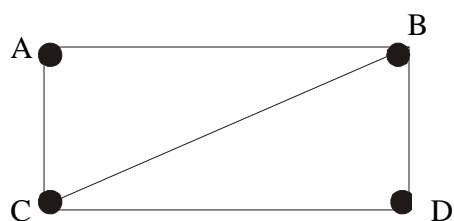
### Trayectoria de Euler:

a) Si una gráfica  $G$  tiene mas de dos vértices de grado impar, entonces no puede existir una trayectoria de Euler en  $G$ .



vértice	grado
A	3
B	3
C	3
D	3
E	4

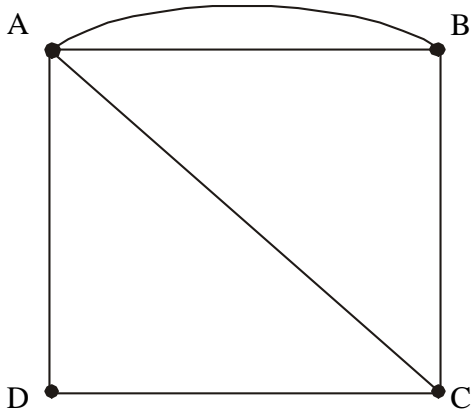
b) Si  $G$  es conexa y tiene exactamente dos vértices de grado impar, entonces existe una trayectoria de Euler en  $G$ . Cualquier trayectoria de Euler debe empezar en un vértice de grado impar y terminar en el otro.



vértice	grado
A	2
B	3
C	3
D	2

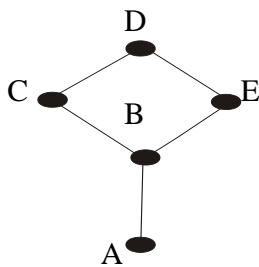
### Trayectoria y circuito Hamiltoniano

**Circuito Hamiltoniano:** Es aquel que tiene cada vértice sólo una vez, excepto el primer vértice, que también es el último.



- La trayectoria A, D, C, B, A (puede elegirse cualquier arista de B a A) es un circuito Hamiltoniano.

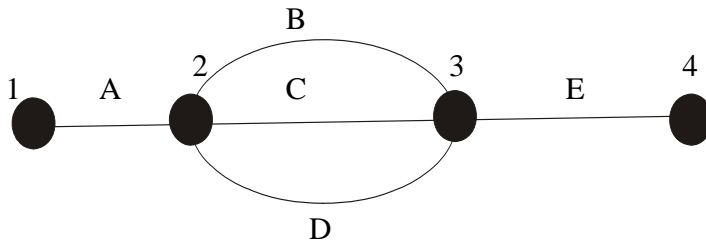
**Trayectoria hamiltoniana:** Es aquella que contiene cada vértice solo una vez.



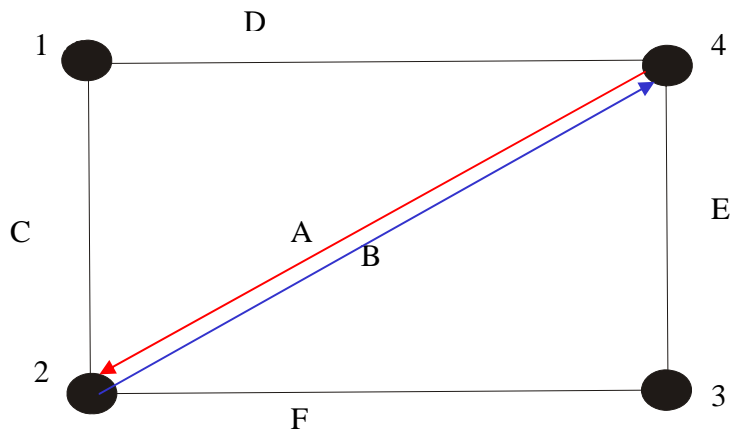
- La trayectoria A, B, C, D, E es una trayectoria Hamiltoniana, pues contiene cada vértice una sola vez.

**Ramas paralelas**

Se le llaman ramas paralelas a los arcos que parten de un mismo vértice o nodo y llegan todos los arcos a un mismo vértice o nodo.



En ejemplo anterior se tiene que los arcos (B, C, D) forman ramas paralelas ya que parten de un mismo nodo y llegan a un mismo nodo.



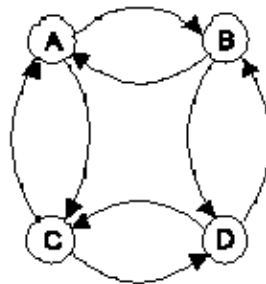
Este ejemplo no puede tener ramas paralelas debido a que los arcos (A, B) no parten de un mismo nodo.

**Matriz de adyacencia.**

Los grafos se representan en memoria secuencial mediante matrices de adyacencia.

Una matriz de adyacencia, es una matriz de dimensión  $n \bullet n$ , en donde  $n$  es el número de vértices que almacena valores booleanos, donde matriz  $M[i, j]$  es verdadero si y solo si existe un arco que vaya del vértice  $i$  y al vértice  $j$ .

Veamos el siguiente grafo dirigido:



La matriz de adyacencia, que se obtuvo a partir del grafo anterior es la siguiente:

$$M = \begin{matrix} & \begin{matrix} A & B & C & D \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

Matriz de adyacencia.

**Grafos dirigidos.**

$G = (V, A)$  un grafo dirigido con  $|V| = n$ . Se define la matriz de adyacencia o booleana asociada a  $G$  como  $B_{n \bullet n}$  con

$$b_{i,j} = \begin{cases} 1 & \text{si } (i,j) \text{ pertenece a } A \\ 0 & \text{en otro caso} \end{cases}$$

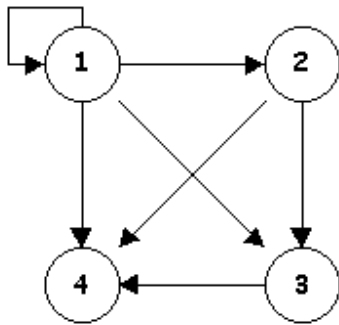
Como se ve, se asocia cada fila y cada columna a un vértice y los elementos  $b_{i,j}$  de la matriz son 1 si existe el arco  $(i,j)$  y 0 en caso contrario.

**Grafos no dirigidos.**

$G=(V,A)$  un grafo no dirigido con  $|V|=n$ . Se define la matriz de adyacencia o booleana asociada a  $G$  como  $B_{n \times n}$  con:

$$b_{ij} = b_{ji} = \begin{cases} 1 & \text{si } (i,j) \text{ pertenece a } A. \\ 0 & \text{en otro caso.} \end{cases}$$

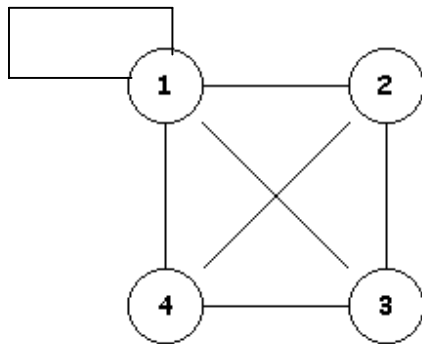
La matriz  $B$  es simétrica con 1 en las posiciones  $ij$  y  $ji$  si existe la arista  $(i,j)$ .

**EJEMPLO:**

	1	2	3	4
1	1	1	1	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

Matriz de Adyacencia del Grafo.

Si el grafo es etiquetado, entonces tanto  $b_{i,j}$  como  $b_{j,i}$  representan al coste o valor asociado al arco  $(i,j)$  y se suelen denominar matrices de coste. Si el arco  $(i,j)$  no pertenece a  $A$  entonces se asigna  $b_{i,j}$  o  $b_{j,i}$  un valor que no puede ser utilizado como una etiqueta válida.



	1	2	3	4
1	1	1	1	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

Matriz de Adyacencia del Grafo.

La principal ventaja de la matriz de adyacencia es que el orden de eficiencia de las operaciones de obtención de etiqueta de un arco o ver si dos vértices están conectados son independientes del número de vértices y de arcos. Por el contrario, existen dos grandes inconvenientes:

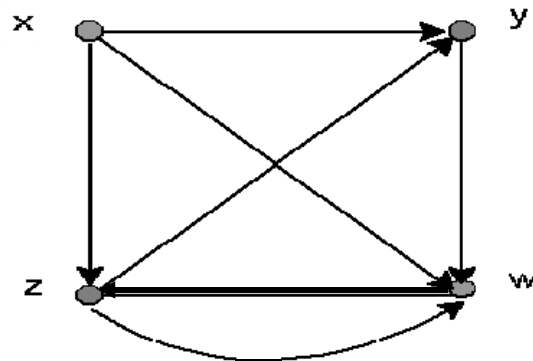
- Es una representación orientada hacia grafos que no modifica el número de sus vértices ya que una matriz no permite que se le o supriman filas o columnas.
- Se puede producir un gran derroche de memoria en grafos poco densos (con gran número de vértices y escaso número de arcos).

Para evitar estos inconvenientes se introduce otra representación: las listas de adyacencia.

**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 **u1 = X**, **u2 = Y**, **u3 = Z**, y **u4 = 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  $a_{ij} = 1$  si hay una arista  $u_i$  a  $u_j$ ; si no  $a_{ij} = 0$ .

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  $B_4 = A + A^2 + A^3 + A^4$ :

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

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

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.

## Isomorfismo

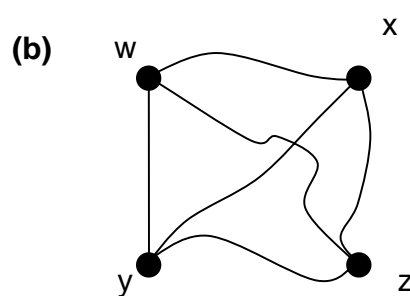
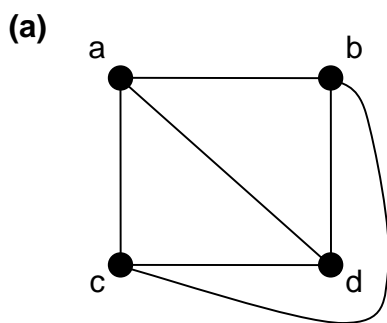
*Definición:*

Sean  $G_1 = (V_1, E_1)$  y  $G_2 = (V_2, E_2)$  dos grafos no dirigidos. Una función  $f: V_1 \rightarrow V_2$  es un isomorfismo de grafos si (a)  $f$  es inyectiva y sobre y (b) para todos  $a, b \in V_1$   $\{a, b\} \in E_1$  si y sólo si  $\{f(a), f(b)\} \in E_2$ . Cuando existe tal función,  $G_1$  y  $G_2$  son grafos isomorfos.

La correspondencia de vértices de un isomorfismo de grafos mantiene las adyacencias. Puesto que el hecho de que los pares de vértices sean adyacentes o no es la única propiedad esencial de un grafo no dirigido, de esta forma preservamos la estructura de los grafos.

Para los siguientes grafos la función  $f$  definida por:

$$f(a) = w, \quad f(b) = x, \quad f(c) = y, \quad f(d) = z$$



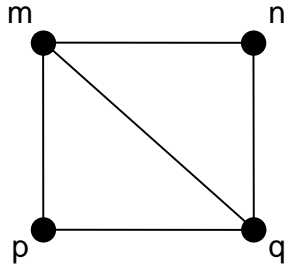
Da como resultado un isomorfismo. De hecho, cualquier correspondencia uno a uno entre  $\{a, b, c, d\}$  y  $\{w, x, y, z\}$  será un isomorfismo, ya que ambos grafos son completos. También esto será cierto si cada uno de los grafos dados tiene solamente cuatro vértices aislados.



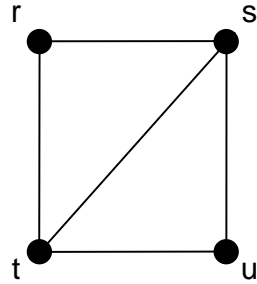
Para los siguientes grafos se necesita ser más cuidadoso. La función  $g$  definida por

$$g(m) = r, \quad g(n) = s, \quad g(p) = t, \quad g(q) = u$$

(c)



(d)



Observemos que aunque  $\{m, q\}$  es una arista del grafo de la parte (c),  $\{g(m), g(q)\} = \{r, u\}$  no es una arista del grafo de la parte (d). En consecuencia, la función  $g$  *no* define un isomorfismo de grafos. Para mantener la correspondencia de aristas se considera la función uno a uno y sobre  $h$  tal que

$$h(m) = s, \quad h(n) = r, \quad h(p) = u, \quad h(q) = t.$$

En este caso tenemos la correspondencia de aristas

$$\{m, n\} \leftrightarrow \{h(m), h(n)\} = \{s, r\}$$

$$\{m, p\} \leftrightarrow \{h(m), h(p)\} = \{s, u\}$$

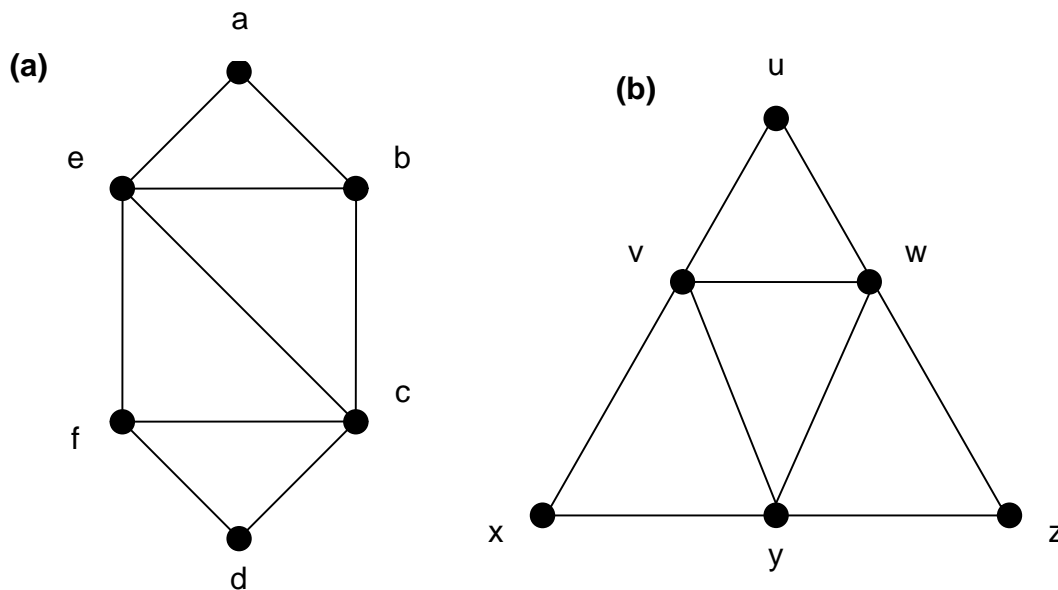
$$\{m, q\} \leftrightarrow \{h(m), h(q)\} = \{s, t\}$$

$$\{n, q\} \leftrightarrow \{h(n), h(q)\} = \{r, t\}$$

$$\{p, q\} \leftrightarrow \{h(p), h(q)\} = \{u, t\}$$

Entonces  $h$  es un isomorfismo de grafos. Notemos también el ciclo:  $m \rightarrow n \rightarrow q \rightarrow m$  corresponde con el ciclo  $s(=h(m)) \rightarrow r(=h(n)) \rightarrow t(=h(q)) \rightarrow s(=h(m))$ .

Ahora analicemos los siguientes dos grafos:



En el grafo (a), el vértice  $a$  es adyacente a otros dos vértices del grafo. En consecuencia, si intentamos construir un isomorfismo entre estos grafos, deberíamos asociar el vértice  $a$  con otro vértice análogo del grafo (b), por ejemplo, el vértice  $u$ . Una situación similar existe para el vértice  $d$  con los dos vértices  $x$  o  $z$ . Pero, independientemente del vértice  $x$  o  $z$  que se use, quedará un vértice en el grafo (b) que es adyacente a otros 2. Y no hay otro vértice en el grafo (a) para continuar con la correspondencia uno a uno que se preserve a la estructura. En consecuencia, estos dos grafos **no son isomorfos**.

### ¿Qué es un árbol y donde se aplican?

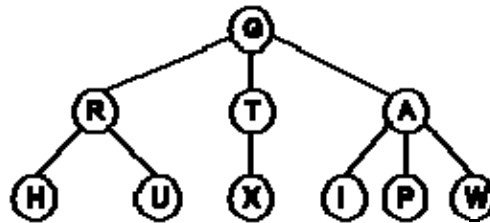
Los árboles representan las estructuras no lineales y dinámicas de datos más importantes en computación. Dinámicas porque las estructuras de árbol pueden cambiar durante la ejecución de un programa. No lineales, puesto que a cada elemento del árbol pueden seguirle varios elementos.

Los árboles pueden ser contruidos con estructuras estáticas y dinámicas. Las estáticas son arreglos, registros y conjuntos, mientras que las dinámicas están representadas por listas.

La definición de árbol es la siguiente: *es una estructura jerárquica aplicada sobre una colección de elementos u objetos llamados nodos; uno de los cuales es conocido como raíz*. Además se crea una relación o parentesco entre los nodos dando lugar a términos como padre, hijo, hermano, antecesor, sucesor, ancestro, etc.

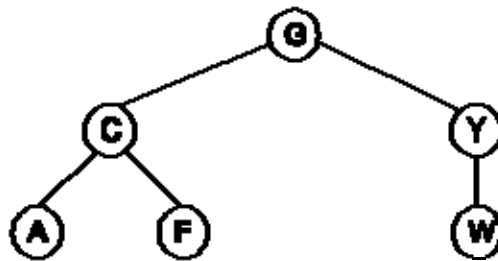
Formalmente se define un árbol de tipo  $T$  como una estructura homogénea que es la concatenación de un elemento de tipo  $T$  junto con un número finito de árboles disjuntos, llamados subárboles. Una forma particular de árbol puede ser la estructura vacía.

La figura siguiente representa a un árbol general.



Se utiliza la recursión para definir un árbol porque representa la forma más apropiada y porque además es una característica inherente de los mismos.

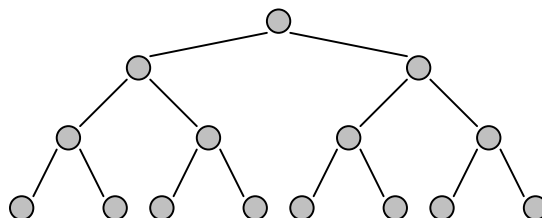
La representación gráfica de un árbol binario es la siguiente:



Ejemplo de aplicaciones de árboles

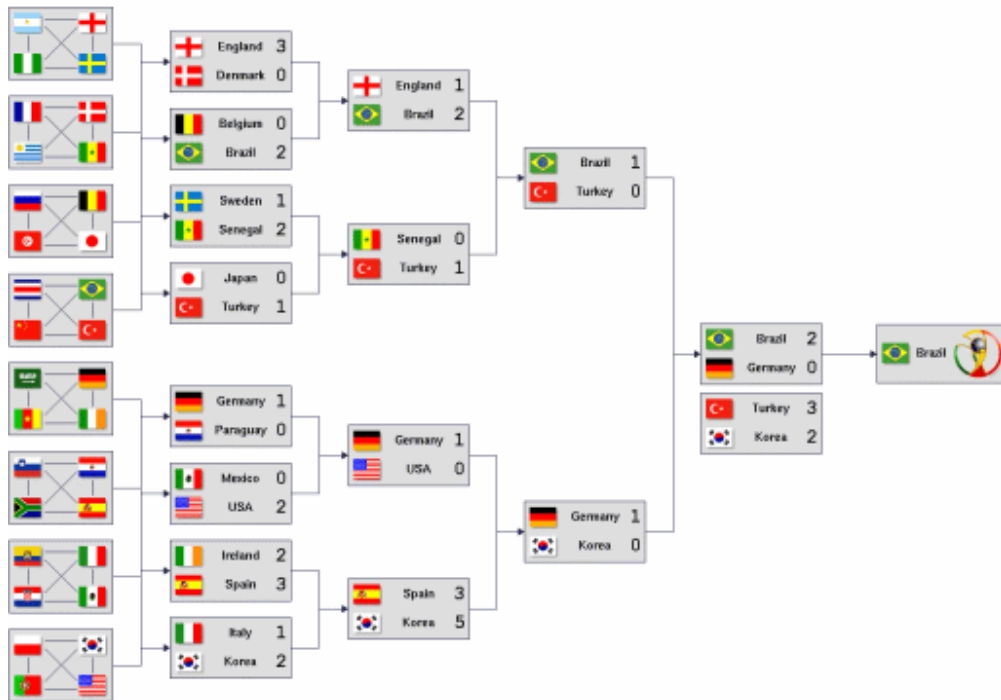
Los árboles tienen una gran variedad de aplicaciones. Por ejemplo, se pueden utilizar para representar fórmulas matemáticas, para organizar adecuadamente la información, para construir un árbol genealógico, en la toma de decisiones, para el análisis de circuitos eléctricos y para numerar los capítulos y secciones de un libro.

A los árboles ordenados de grado dos se les conocen como árboles binarios ya que cada nodo del árbol no tendrá más de dos descendientes directos. Las aplicaciones de los árboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual es posible tomar decisiones con dos opciones en distintos puntos.

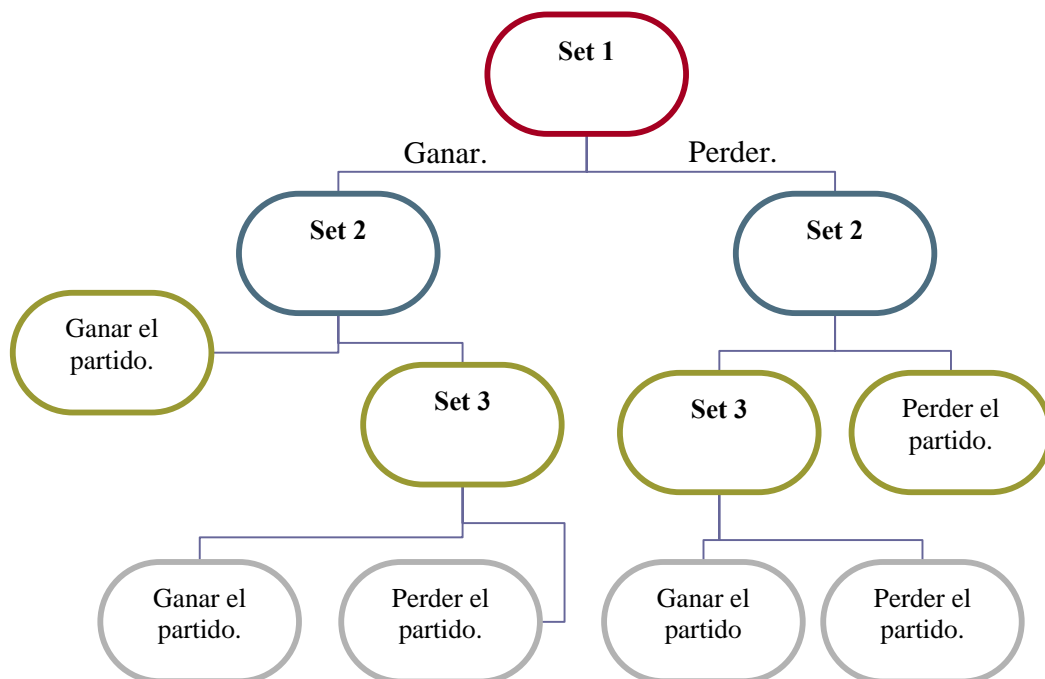


A continuación podremos observar un caso típico del uso de árboles.

### Ejemplo de un árbol



### Partido de tenis a “2 o 3 sets.”



En ciencias de la computación, un árbol es una estructura de datos comúnmente usada que emula la estructura de un árbol con un conjunto de nodos conectados.

Cada nodo tiene cero o más nodos hijos, que están por debajo de él (en ciencias de la computación, al contrario que en la naturaleza, los árboles crecen hacia abajo, no hacia arriba), El nodo del cual un nodo es hijo es llamado su nodo padre. Un hijo tiene como máximo un padre; un nodo sin padre es llamado nodo raíz (o simplemente raíz). Los nodos sin hijos son llamados hojas.

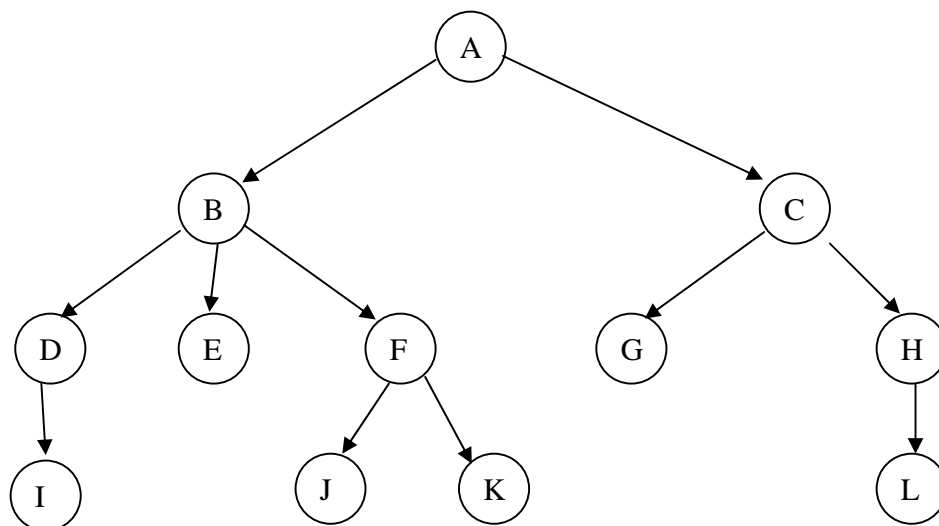
En teoría de grafos, un árbol es un digrafo conectado acíclico. Un árbol con raíz es como un grafo con un vértice seleccionado como la raíz. En ese caso, sólo dos vértices conectados con el lado heredan una relación de padre-hijo. *Un grafo acíclico con múltiples componentes conectados o un conjunto de árboles con raíz se llama bosque.*

### Características y propiedades de los árboles.

Las siguientes son las características y propiedades más importantes de los árboles en general:

- Todo árbol que no es vacío, tiene un único nodo raíz.
- Un nodo X es descendiente directo de un nodo Y, si el nodo X es apuntado por el nodo Y. en este caso es común utilizar la expresión **X es hijo de Y**.
- Un nodo X es antecesor directo de un nodo Y, si el nodo X apunta al nodo Y. en es caso es común utilizar la expresión **X es padre de Y**.
- Se dice que todos los nodos que son descendientes directos (hijos) de un mismo nodo (padre), son **hermanos**.
- Todo nodo que no tiene ramificaciones (hijos), se conoce con el nombre de **terminal** u **hoja**.
- Todo nodo que no es raíz, ni terminal u hoja se conoce con el nombre de **interior**.
- Grado** es el número de descendientes directos de un determinado nodo. **Grado del árbol** es el máximo grado de todos los nodos del árbol, es decir, el grado más alto entre todos los nodos.
- Nivel** es el número de arcos que deben ser recorridos para llegar a un determinado nodo. Por definición la raíz tiene nivel 1.
- Altura** del árbol es el máximo número de niveles de todos los nodos del árbol.

A continuación se presenta un ejemplo para clarificar estos conceptos.



1. A es la raíz del árbol.
2. B es hijo de A.  
C es hijo de A.  
D es hijo de B.  
E es hijo de B.  
L es hijo de H.
3. A es padre de B.  
B es padre de D.  
D es padre de I.  
C es padre de G.  
H es padre de L.
4. B y C son hermanos.  
D, E y F son hermanos.  
G y H son hermanos.  
J y K son hermanos.
5. I, E, J, K, G y L son nodos terminales u hojas.
6. B, D, F, C y H son nodos interiores.
7. El grado del nodo A es 2.  
El grado del nodo B es 3.  
El grado del nodo C es 2.  
El grado del nodo D es 1.  
El grado del nodo E es 0.  
  
El grado del árbol es 3.
8. El nivel del nodo A es 1.  
El nivel del nodo B es 2.  
El nivel del nodo D es 3.  
El nivel del nodo C es 2.  
El nivel del nodo L es 4.
9. La altura del árbol es 4.

### Longitud de camino.

Se define la longitud de camino X como el número de arcos que deben ser recorridos para llegar desde la raíz al nodo X. Por definición la raíz tiene longitud de camino 1, sus descendientes directos tiene longitud de camino 2 y así sucesivamente. En la figura anterior, el nodo B tiene longitud de camino 2, el nodo I longitud de camino 4 y el nodo H longitud de camino 3.

### Longitud de camino interno.

La longitud de camino interno es la suma de las longitudes de camino de todos los nodos del árbol. Puede calcularse por medio de la siguiente fórmula.

$$LCI = \sum_{i=1}^h n_i * i$$

donde i representa el nivel del árbol, h su altura y  $n_i$  el número de nodos en el nivel i.

La LCI del árbol de la figura anterior se calcula así:

$$LCI = 1 * 1 + 2 * 2 + 5 * 3 + 4 * 4 = 36$$

Ahora bien; la media de la longitud de camino interno (LCIM) se calcula dividiendo la LCI entre el número de nodos del árbol (n). Se expresa:

$$LCIM = LCI / n$$

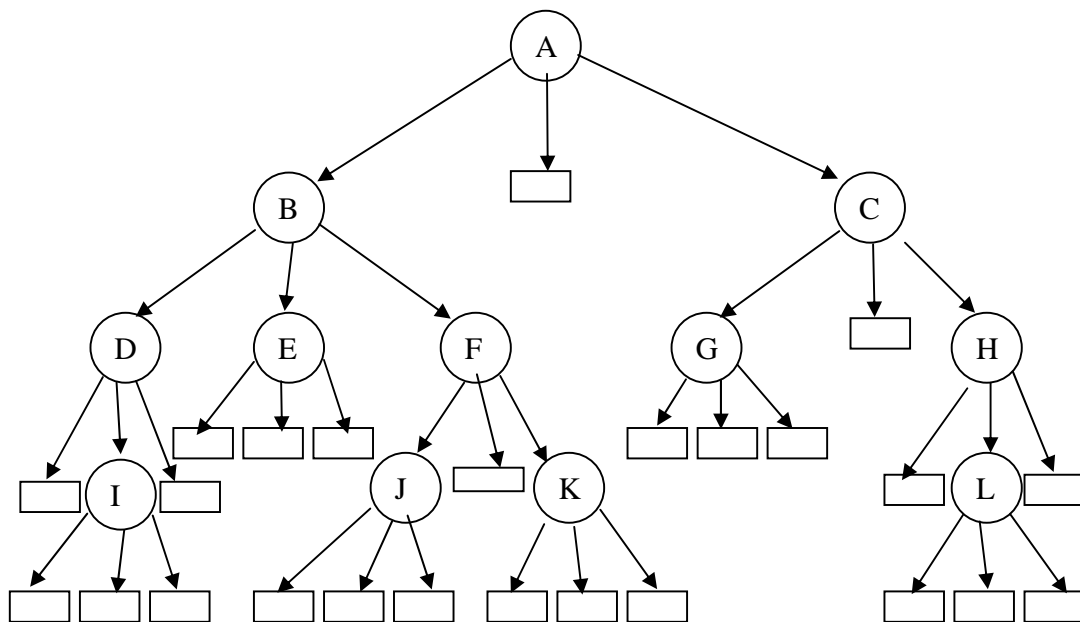
y significa el número de arcos que deben ser recorridos en promedio para llegar partiendo desde la raíz, a un nodo cualquiera del árbol.

$$LCIM = 36 / 12 = 3$$

### Longitud de camino externo.

Para definir la longitud de camino externo, es necesario primero definir los conceptos **árbol extendido** y **nodo especial**. Un árbol extendido es aquel en el que el número de hijos de cada nodo es igual al grado del árbol. Si alguno de los nodos del árbol no cumple con esta condición, entonces deberá incorporársele al mismo nodos especiales; tantos como sean necesarios para satisfacer la condición. Los nodos especiales tiene como objetivo remplazar las ramas vacías o nulas., no pueden tener descendientes y normalmente se representan con la forma de un cuadrado.

A continuación representemos el mismo árbol en forma extendida.



El número de nodos especiales de este árbol es 25.

Se puede definir ahora la longitud de camino externo como la suma de las longitudes de camino de todos los nodos especiales del árbol. Se calcula por medio de la siguiente fórmula:

$$LCE = \sum_{i=2}^{h+1} ne_i * i$$

Donde  $i$  representa el nivel del árbol,  $h$  su altura y  $ne_i$  el número de nodos especiales en el nivel  $i$ . Obsérvese que  $i$  comienza desde 2, puesto que la raíz se encuentra en el nivel 1 y no puede ser un nodo especial.

La LCE del árbol de la figura anterior se calcula así:

$$LCE = 1 * 2 + 1 * 3 + 11 * 4 + 12 * 5 = 109$$

Ahora bien; la media de la longitud de camino externo (LCEM) se calcula dividiendo la LCE entre el número de nodos especiales del árbol ( $ne$ ). Se expresa:

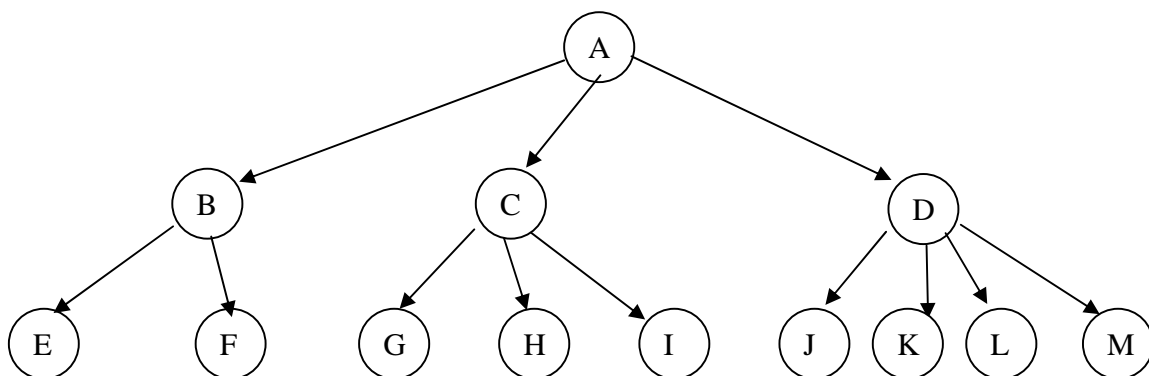
$$LCEM = LCE / ne$$

y significa el número de arcos que deben ser recorridos en promedio para llegar partiendo desde la raíz, a un nodo especial cualquiera del árbol.

$$LCEM = 109 / 25 = 4.36$$

Un ejemplo clarificará los conceptos de longitud interno y externo.

Ejemplo: Dado el árbol general y el árbol extendido de las siguientes figuras, se calcula:



Árbol general

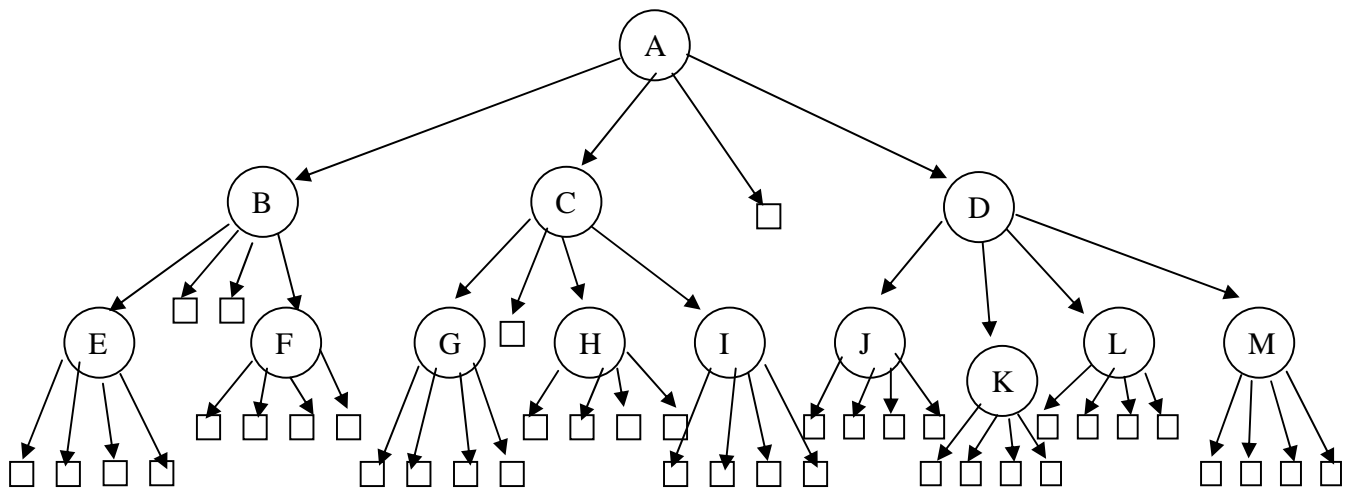
La longitud de camino interno:

$$LCI = 1 * 1 + 3 * 2 + 9 * 3 = 34$$

La media de la longitud de camino interno:

$$LCIM = 34 / 13 = 2.61$$





Árbol extendido

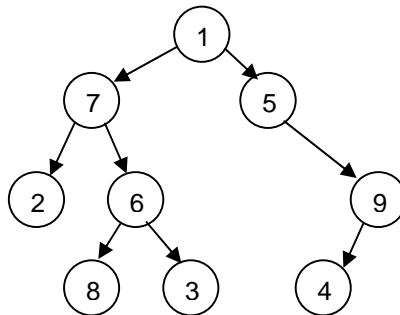
La longitud de camino externo:

$$\text{LCE} = 1 * 2 + 3 * 3 + 36 * 4 = 155$$

La media de la longitud de camino externo:

$$\text{LCEM} = 155 / 40 = 3.87$$

En una estructura de datos en árbol, no hay distinción entre los diferentes hijos de un nodo, ninguno es el primer hijo o el último hijo. Un árbol en el que se distinguen los hijos por su orden se llama árbol ordenado, y las estructuras de datos construidas en ellos se llaman estructuras de datos de árboles ordenados.



Un ejemplo sencillo de un árbol binario

Un Árbol binario es un tipo de árbol ordenado, y hay una relación uno a uno entre los árboles binarios y los árboles ordenados generales

Hay muchas formas diferentes de representar un árbol, representaciones comunes muestran los nodos como variables localizadas en el montículo como punteros a sus hijos, sus padres, o juntos, o como elementos de un array (arreglo), con relaciones determinadas entre posiciones en el arreglo (ej., heap binario).

Ir entre los elementos de un árbol, por medio de las conexiones entre padres e hijos, es llamado **Caminar por el árbol**, y la acción es **caminata** por el árbol. A menudo, una operación puede ser

realizada cuando un apuntador arriba a un nodo en particular. Una caminata donde las operaciones suceden a un nodo antes de suceder a sus hijos es llamada una **primera caminata ancha**; una caminata donde los hijos son operados antes de que el padre sea llamado se llama **primera caminata profunda**.

Véase también: Árbol Binario para una discusión de pre-orden, en-orden y post-orden transversal

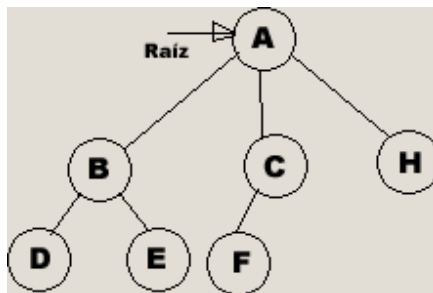
Operaciones comunes en árboles son:

- Enumerar todos los elementos
- Buscar un elemento
- Añadir un nuevo ítem en una cierta posición del árbol
- Borrar un elemento
- Eliminar una sección completa de un árbol (llamado **podar**);
- Añadir una sección completa a un árbol (llamado **injertar**);
- Encontrar la raíz de cualquier nodo

## ÁRBOLES BINARIOS DE BÚSQUEDA

### ¿Qué son los árboles binarios de búsqueda?

Empezaremos recordando lo que son los árboles. Un árbol es una colección de nodos que puede estar vacía o no.



Si no está vacía, el árbol estará formado por un nodo raíz y cero o más subárboles que están unidos a la raíz por otras tantas aristas.

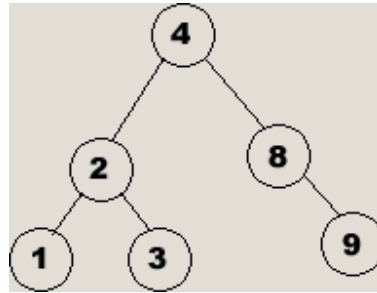
Para que un árbol sea binario es requisito indispensable el que el número máximo de hijos que tenga cada nodo sea 2. Por lo tanto el árbol anterior no es un árbol binario, ya que el nodo A tiene 3 hijos.

Por último, un árbol será de búsqueda si todos sus nodos cumplen las siguientes condiciones:

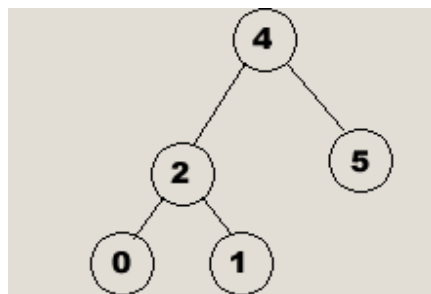
- Todos los nodos situados a su izquierda son menores que él.
- Todos los nodos situados a su derecha son mayores que él.

Resumiendo, se puede decir que un árbol binario de búsqueda es un árbol en el que cada nodo tiene a lo sumo dos hijos, y en el que para cada nodo todos los nodos a su izquierda son menores que él y todos los nodos a su derecha son mayores que él.

El siguiente árbol es un ejemplo de árbol de búsqueda:



En cambio este árbol viola la condición de orden en el nodo 2, ya que un nodo a su derecha, 1, no es mayor que él.



### Transformación de un árbol general en un árbol binario.

En esta sección estableceremos los mecanismos necesarios para convertir un árbol general en un árbol binario. Para esto, debemos seguir los pasos que se describen a continuación:

1. Enlazar los hijos de cada nodo en forma horizontal (los hermanos).
2. Enlazar en forma vertical el nodo padre con el nodo hijo que se encuentra más a la izquierda. Además, debe eliminarse el vínculo de ese padre con el resto de sus hijos.
3. Rotar el diagrama resultante aproximadamente 45 grados hacia la izquierda, y así se obtendrá el árbol binario correspondiente.

## OPERACIONES BÁSICAS EN ÁRBOLES BINARIOS DE BÚSQUEDA

Como en toda estructura de datos hay dos operaciones básicas, inserción y eliminación.

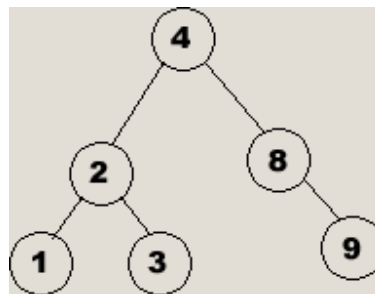
**Inserción:** El procedimiento de inserción en un árbol binario de búsqueda es muy sencillo, únicamente hay que tener cuidado de no romper la estructura ni el orden del árbol.

Cuando se inserta un nuevo nodo en el árbol hay que tener en cuenta que cada nodo no puede tener más de dos hijos, por esta razón si un nodo ya tiene 2 hijos, el nuevo nodo nunca se podrá insertar como su hijo. Con esta restricción nos aseguramos mantener la estructura del árbol, pero aún nos falta mantener el orden.

Para localizar el lugar adecuado del árbol donde insertar el nuevo nodo se realizan comparaciones entre los nodos del árbol y el elemento a insertar. El primer nodo que se compara es la raíz, si el nuevo nodo es menor que la raíz, la búsqueda prosigue por el nodo izquierdo de éste. Si el nuevo nodo fuese mayor, la búsqueda seguiría por el hijo derecho de la raíz.

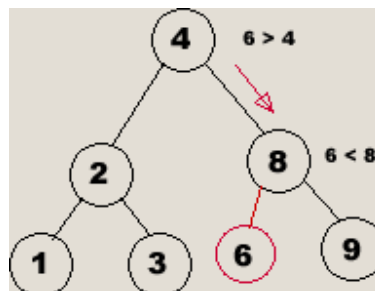
Este procedimiento es recursivo, y su condición de parada es llegar a un nodo que no tenga hijo en la rama por la que la búsqueda debería seguir. En este caso el nuevo nodo se inserta en ese hueco, como su nuevo hijo.

Vamos a verlo con un ejemplo sobre el siguiente árbol:



Se quiere insertar el elemento 6.

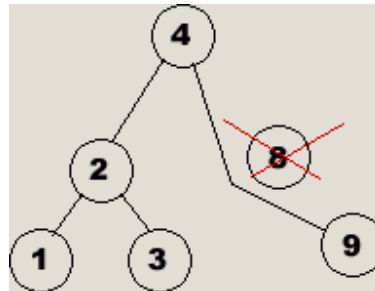
Lo primero es comparar el nuevo elemento con la raíz. Como  $6 > 4$ , entonces la búsqueda prosigue por el lado derecho. Ahora el nuevo nodo se compara con el elemento 8. En este caso  $6 < 8$ , por lo que hay que continuar la búsqueda por la rama izquierda. Como la rama izquierda de 8 no tiene ningún nodo, se cumple la condición de parada de la recursividad y se inserta en ese lugar el nuevo nodo.



**Borrar:** El borrado en árboles binarios de búsqueda es otra operación bastante sencilla excepto en un caso. Vamos a ir estudiando los distintos casos.

Tras realizar la búsqueda del nodo a eliminar observamos que el nodo no tiene hijos. Este es el caso más sencillo, únicamente habrá que borrar el elemento y ya habremos concluido la operación.

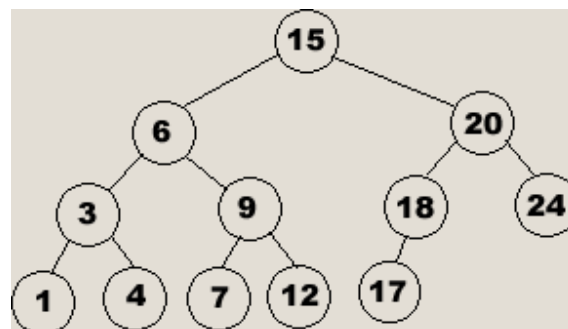
Si tras realizar la búsqueda nos encontramos con que tiene un sólo hijo. Este caso también es sencillo, para borrar el nodo deseado, hacemos una especie de *punte*, el padre del nodo a borrar pasa a apuntar al hijo del nodo borrado.



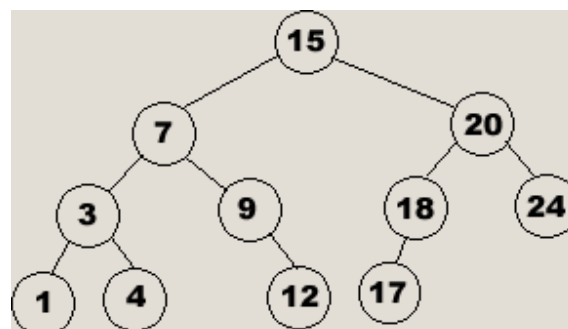
Por último, el caso más complejo, si el nodo a borrar tiene dos hijos. En este caso se debe sustituir el nodo a borrar por mayor de los nodos menores del nodo borrado, o por el menor de los nodos mayores de dicho nodo. Una vez realizada esta sustitución se borra el nodo que sustituyó al nodo eliminado (operación sencilla ya que este nodo tendrá un hijo a lo sumo).

Sobre el siguiente árbol queremos eliminar el elemento 6. Tenemos dos opciones para sustituirlo:

- El menor de sus mayores: 7.
- El mayor de sus menores: 4.
- 



Vamos a sustituirlo por el 7 (por ejemplo). El árbol resultante sería el siguiente, tras eliminar también el elemento 7 de su ubicación original.



### Otras operaciones

En los árboles de búsqueda la operación buscar es muy eficiente. El algoritmo compara el elemento a buscar con la raíz, si es menor continua la búsqueda por la rama izquierda, si es mayor continua por la derecha. Este procedimiento se realiza recursivamente hasta que se encuentra el nodo o hasta que se llega al final del árbol.

Otra operación importante en el árbol es el recorrido el mismo. El recorrido se puede realizar de tres formas diferentes:

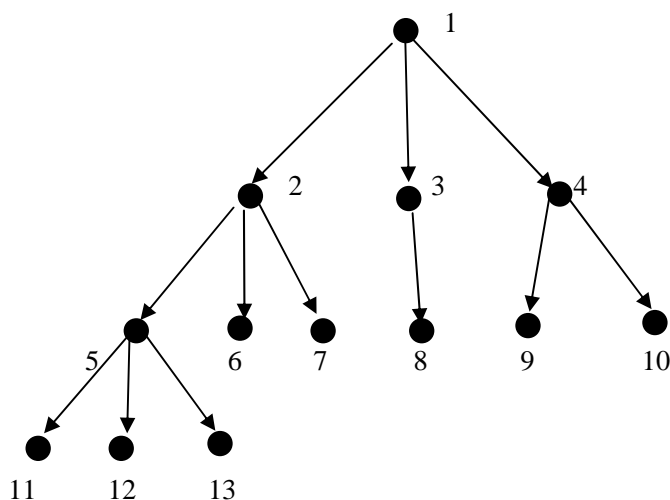
- **Preorden:** Primero el nodo raíz, luego el subárbol izquierdo y a continuación el subárbol derecho.
- **In orden:** Primero el subárbol izquierdo, luego la raíz y a continuación el subárbol derecho.
- **Postorden:** Primero el subárbol izquierdo, luego el subárbol derecho y a continuación la raíz.

### Utilización de árboles binarios de búsqueda.

Los árboles binarios de búsqueda son unas estructuras de datos muy utilizadas en la informática. Una de sus más importantes aplicaciones es el almacenamiento de información asociada con claves de búsqueda, ya que permite un almacenamiento y recuperación de la información muy eficiente.

Sean  $T$  un árbol ordenado y  $A$  el conjunto de todos los vértices de  $T$ . Se define un **árbol** posicional binario  $B(t)$  sobre el conjunto de vértices  $A$ , como sigue. Si  $v \in A$ , entonces, el hijo izquierdo  $V_L$  de  $v$  en  $B(t)$  es el primer hijo de  $v$  en  $T$ , si éste existe. El hijo derecho  $V_R$  de  $v$  en  $B(t)$  es el siguiente hermano de  $v$  en  $T$  (en el orden dado de los hermanos de  $T$ ), si es que este existe.

**Ejemplo.** La figura a continuación muestra el dígrafo de un árbol etiquetado  $T$ . Se supone que cada conjunto de hermanos está ordenado de izquierda a derecha, como en el dibujo. Así, los hijos del vértice 1, es decir, los vértices 2, 3 Y 4, quedan ordenados con el vértice 2 en primer lugar, el 3, en segundo y el 4, en tercero. De manera similar, el primer hijo del vértice 5 es el vértice 11, el segundo es el vértice 12 y el tercero es el vértice 13.

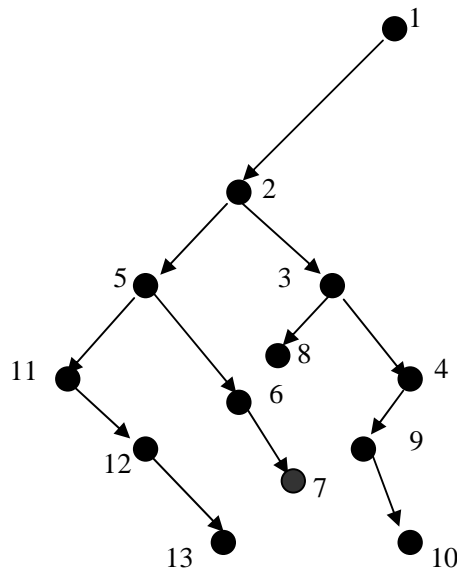


En la siguiente figura aparece el dígrafo del árbol posicional binario correspondiente,  $B(T)$ .

Para obtenerla basta trazar una arista izquierda desde cada uno de los vértices  $v$  hacia su primer hijo (si tiene hijos). Después se traza una arista derecha de cada uno de sus vértices  $v$  hacia su siguiente hermano.

Así, la arista izquierda del vértice 2, va al vértice 5, ya que el vértice 5 es el primer hijo del vértice 2 en el árbol  $T$ . Además, la arista derecha del vértice 2, va hacia el vértice 3, ya que el vértice 3 es el siguiente hermano en el renglón (entre todos los hijos del vértice 1).

Con frecuencia, la representación de  $B(T)$  a manera de lista doblemente enlazada se llama **representación en lista enlazada de  $T$** .



### ÁRBOLES ETIQUETADOS

A veces es necesario etiquetar los vértices o aristas de un dígrafo para indicar su uso para su propósito específico.

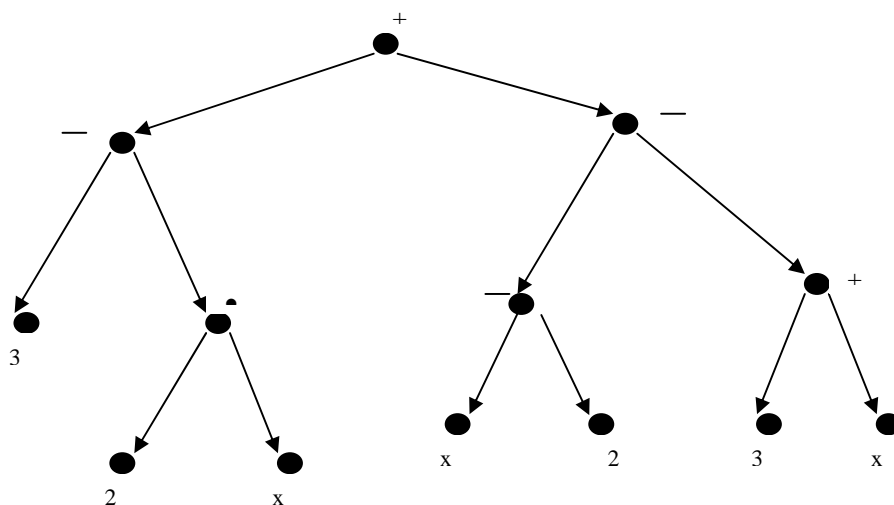
Esto es particularmente cierto para muchos de los árboles es la ciencia de la computación.

Ahora se proporcionará un ejemplo donde el conjunto de los vértices no son importantes, sino que la utilidad del árbol se enfatiza mediante las etiquetas sobre estos vértices.

Así, se representará los vértices como puntos y se mostrará la etiqueta de cada vértice junto al punto que representa dicho vértice.

EJEMPLO:

$$(3 - (2 \bullet x)) + ((x - 2) - (3 + x))$$



En esta expresión se supone que no es posible realizar operaciones como  $-$ ,  $+$ ,  $x$ , o  $/$  hasta evaluar ambos argumentos; Es decir hasta realizar todos los cálculos dentro de los argumentos de la izquierda y de la derecha.

Por lo tanto, no es posible realizar la suma central hasta haber evaluado

$$(3 - (2 \cdot x)) \text{ y } ((x - 2) - (3 + x)).$$

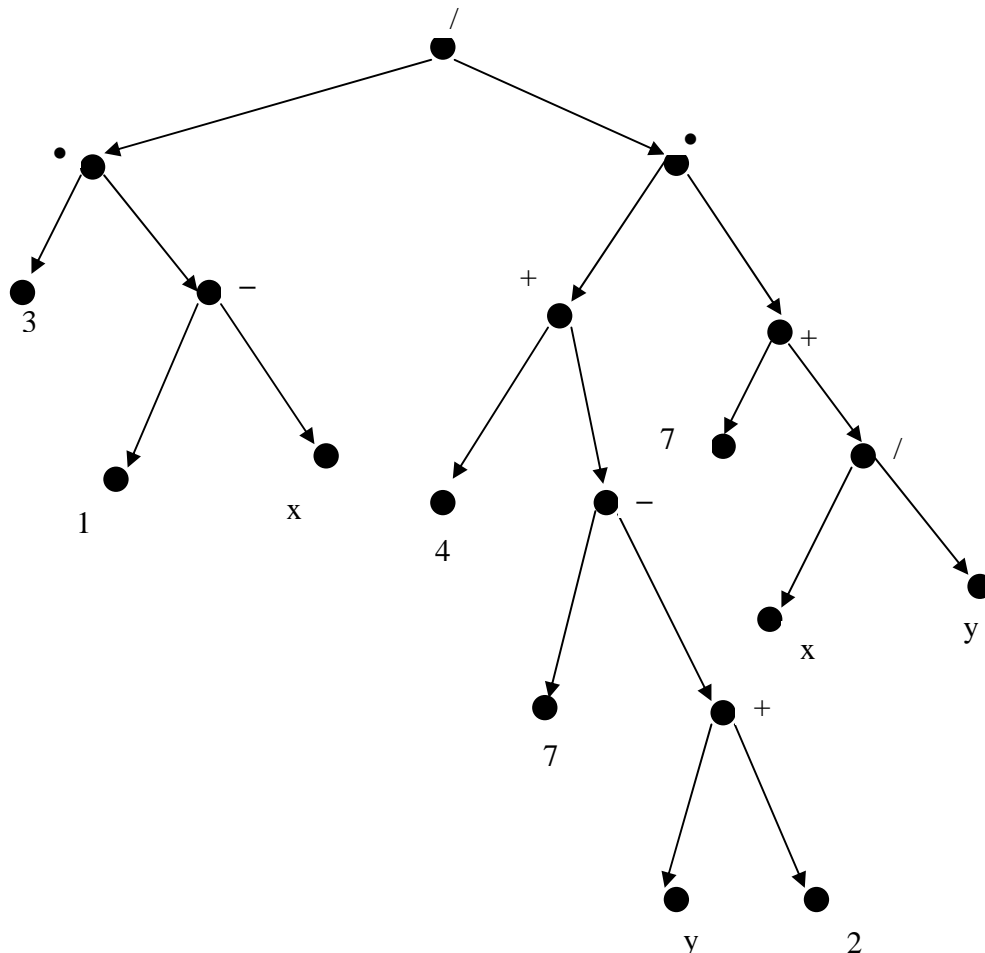
No es posible realizar la resta central en  $((x-2) - (3+x))$ .

Hasta haber evaluado  $((x - 2)$  y  $(3 + x))$ .

Y así sucesivamente.

Este muestra el árbol correspondiente a la expresión con cada operación entre paréntesis.

$$(3 \cdot (1 - x)) / ((4 + (7 - (y + 2))) \cdot (7 + (x / y))).$$





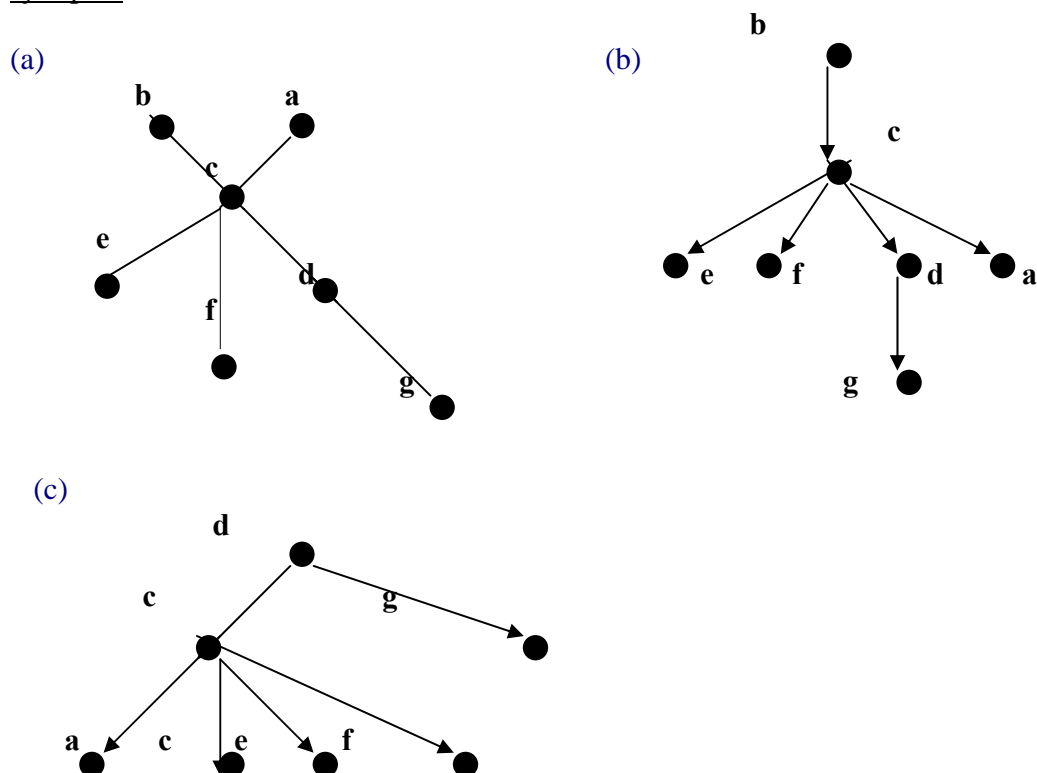
## ÁRBOLES NO DIRIGIDOS

Un árbol no dirigido es la cerradura simétrica de un árbol. Es decir, es un árbol con todas sus aristas bidireccionales. Como se acostumbra con las relaciones simétricas, se representa un árbol no dirigido mediante su gráfica, en vez de su dígrafo. La gráfica de un árbol no dirigido  $T$  tendrá una única línea sin flechas que une los vértices  $a$  y  $b$  siempre que  $(a, b)$  y  $(b, a)$  pertenezcan a  $T$ .

El conjunto  $\{a, b\}$ , donde  $(a, b)$  y  $(b, a)$  están en  $T$ , es una arista no dirigida de  $T$ . En este caso los vértices  $a$  y  $b$  son vértices adyacentes. Así cada arista no dirigida  $\{a, b\}$  corresponde a dos aristas ordinarias  $(a, b)$  y  $(b, a)$ . Las líneas de la gráfica de un árbol no dirigido  $T$  corresponden a las aristas no dirigidas en  $T$ .

**Ejemplo 1.** La figura muestra la gráfica de un árbol no dirigido  $T$ . En la figura (b) y (c) se muestran los dígrafos de los árboles ordinarios  $T_1$  y  $T_2$  respectivamente que tienen a  $T$  como cerradura simétrica. Esto muestra que un árbol no dirigido corresponde en general a muchos árboles dirigidos. Se incluye las etiquetas para mostrar la correspondencia de los vértices subyacentes en las tres relaciones.

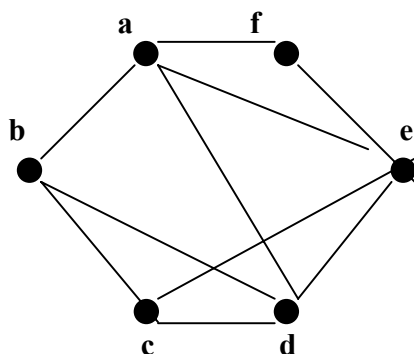
ejemplos



Se requiere presentar algunas definiciones alternativas útiles de un árbol no dirigido, y para esto se necesita algunos comentarios acerca de las relaciones simétricas.

Sea  $R$  una relación simétrica y sea  $p: v_1, v_2, \dots, v_n$  una trayectoria en  $R$ . Se dice que  $p$  es simple si no existen dos aristas de  $p$  correspondientes a la misma arista no dirigida. Sí, además  $v_1$  es igual a  $v_n$  (de modo que  $p$  sea un ciclo),  $p$  es un ciclo simple.

**Ejemplo 2.** La figura muestra la gráfica de una relación simétrica  $R$ . La trayectoria  $a, b, c, e, d$  es simple, pero la trayectoria  $f, e, d, c, d, a$ , no lo es, ya que  $d, c$ , y  $c, d$  corresponden a la misma arista no dirigida. También,  $f, e, a, d, b, a, f$  y  $d, a, b$ , son ciclos simples, pero  $f, e, d, c, e, f$ , no es un ciclo simple, ya que  $f, e$ , y  $e, f$ , corresponden a la misma arista no dirigida.



Una relación simétrica  $R$  es acíclica, si no contiene ciclos simples. Se puede mostrar que si  $R$  contiene ciclos, entonces contiene un ciclo simple. Recuerde que una relación simétrica  $R$  es conexa si existe una trayectoria en  $R$  desde su vértice hacia cualquier otro vértice.

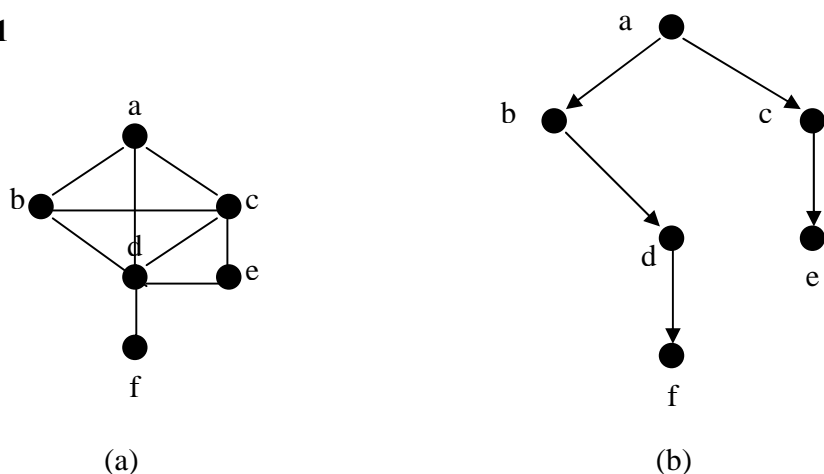
### ÁRBOLES DE EXPANSIÓN EN RELACIONES CONEXAS.

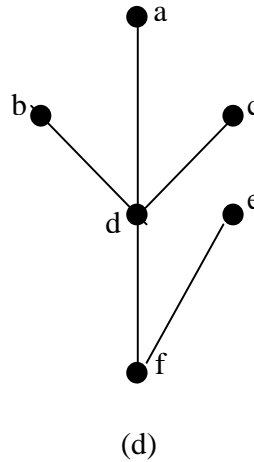
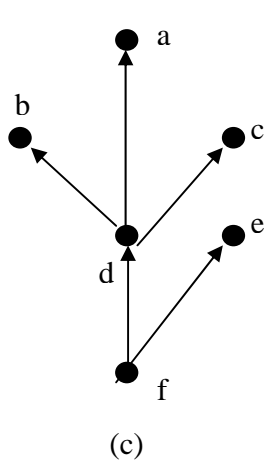
Si la  $R$  es una relación simétrica conexa sobre un conjunto  $A$ , un árbol  $T$  en  $A$  es un árbol de expansión para  $R$  si  $T$  es un árbol con exactamente los mismos vértices que  $R$  y que se puede obtener de  $R$  eliminando algunas aristas de  $R$ .

#### Ejemplo.

La relación simétrica  $R$  cuya gráfica aparece (a) tiene un árbol  $T'$  como árbol de expansión, cuyo dígrafo aparece a continuación (b), también el árbol  $T'$  es un árbol de expansión para  $R$ , cuyo dígrafo es mostrado en la figura D como  $R, T$  y  $T''$  son relaciones en el mismo conjunto  $A$ , se ha etiquetado los vértices para mostrar la correspondencia de los elementos. Como lo muestra este Ejemplo: Los árboles de expansión no son únicos.

#### Ejemplo num. 1

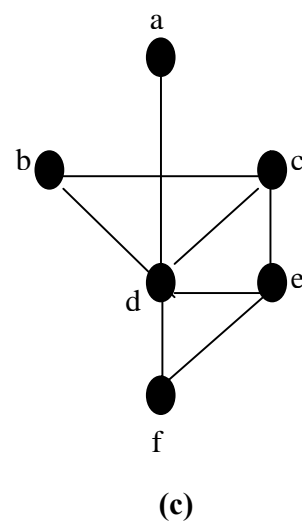
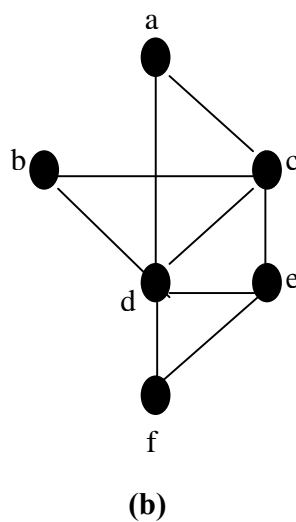
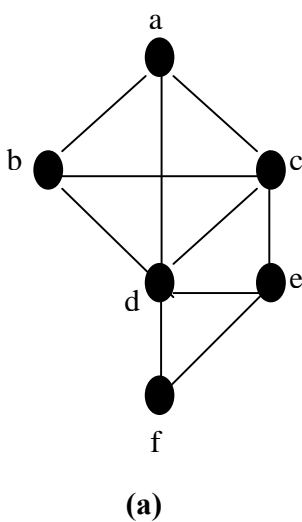


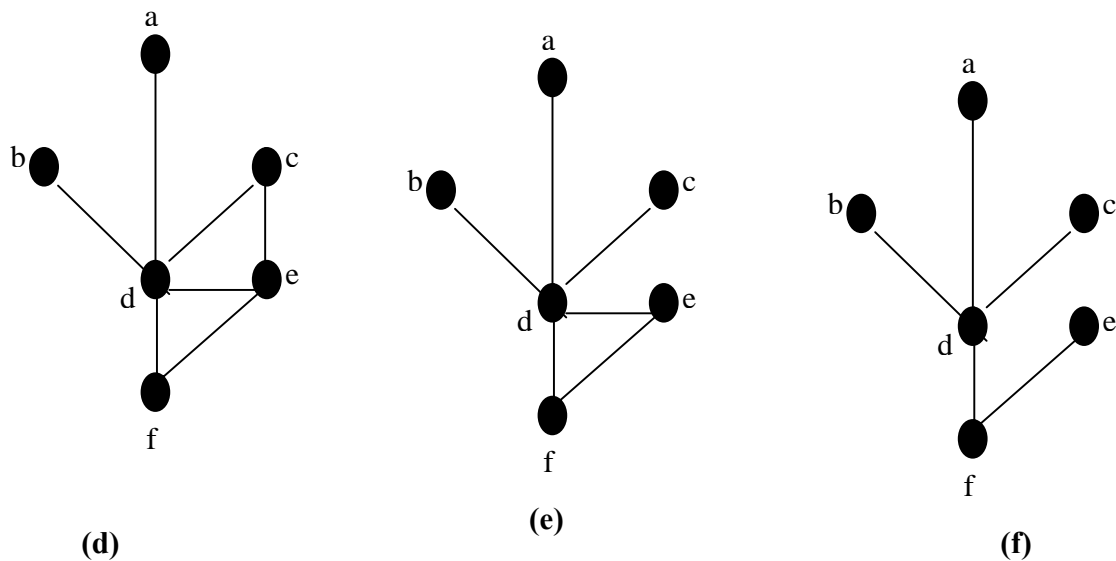


Árboles de expansión no dirigidos. Para una relación simétrica conexa  $R$  este árbol es la cerradura simétrica de un árbol de expansión la figura (d) muestra un árbol de expansión no dirigido para  $R$ , deducido del árbol de expansión de la figura (c) si  $R$  es una relación complicada que es simétrica y conexa, podría ser difícil diseñar un esquema para la búsqueda de  $R$ , para visitar cada uno de sus vértices una vez en cierta forma simétrica. Si  $R$  se reduce a un árbol de expansión, puede utilizarse los algoritmos de búsqueda

**Teorema 2.** Sugiere que un algoritmo para determinar un árbol de expansión no dirigido para una relación  $R$ . Solo hay que eliminar varias aristas no dirigidas de  $R$ . Hasta llegar a un punto donde la eliminación de una o más de las aristas no dirigidas produciría una relación no conexa. El resultado será un árbol de expansión no dirigido.

### Ejemplo num. 2



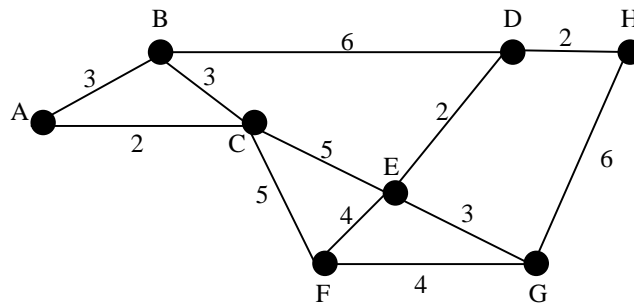


En las figura num. 1 (a) se repite la gráfica en la figura num. 2 (a) se repite se muestra como eliminar varias aristas no dirigidas, lo que culmina en la figura num. 2 (f), el arbol de expansión no dirigido, acorde con la figura num. 2 (d)

Supóngase ahora que se une los vértices  $a$  y  $b$  de una relacion  $R$  en un nuevo vértice  $a'$  que reemplaza a ambos vertices para obtener la relacion  $R'$  para determinar la matriz de  $r'$  se procede la de la siguiente manera.

1. El renglón  $i$  representa al vértice  $a$  y el renglón  $j$  representa al vertice  $b$ . Se reemplaza el renglón  $i$  con la unión de los renglones  $i$  y  $j$  la unión de las dos  $n$ -adas de ceros y unos tiene un  $1$  en una posición presisamente cuando alguna de las dos  $n$ -adas tiene un  $1$  en esa posición.
2. Se reemplaza la columna  $i$  por la unión de las columnas  $i$  y  $j$ .
3. Se restaura la diagonal principal con sus valores originales en  $R$
4. Se elimina el renglón  $j$  y la columna  $j$ .

**Arboles de expansión mínima:** la gráfica no dirigida de la relación modela una situación donde las aristas y los vértices contienen información. Una gráfica con pesos es una gráfica donde cada arista está etiquetada con un valor numérico que denota su peso.



El peso de una arista  $(v_i, v_j)$  es la distancia entre los vértices  $v_i$  y  $v_j$  un vértice  $u$  es un vecino más cercano del vértice  $v$  si  $u$  y  $v$  son adyacentes y ningún otro vértice queda unido con  $v$  mediante una arista de menor peso que  $(u, v)$ . Observe que, a diferencia de lo que sucede en la gramática,  $v$  puede tener más de un vecino cercano.

En las aplicaciones de las gráficas con pesos, con frecuencia se necesita determinar un árbol de expansión no dirigido para el cual el peso total de las aristas en el árbol sea menor .

En las gráficas con pesos, con frecuencia se necesita determinar un arbol de expansión no dirigido para el cual el peso total de las aristas en el árbol sea el menor posible. Este árbol se conoce como árbol de expansión mínima. El algoritmo de Prim se puede adaptar fácilmente para producir un arbol de expansión mínima para la gráfica con pesos. A continuación se enunciará el algoritmo de Prim como si fuera aplicado a una relación simétrica conexa, dada por su gráfica de pesos no dirigida.

#### Algoritmo de Prim.

1. Se elige un vértice  $v_1$  de  $R$ . Sea  $V=\{v_1\}$  y  $E=\{ \}$ .
2. Se elige uno de los vecinos más cercanos a  $v_1$  de  $V$ , que sea adyacente a  $v_j$ ,  $v_j \in V$ , y tal que la arista  $(v_i, v_j)$  no forme un ciclo con miembros de  $E$ . se agrega  $v_i$  a  $V$  y  $(v_i, v_j)$  a  $E$ .
3. Se repite el paso 2 hasta que  $|E| = n - 1$  entonces  $V$  contiene los vértices de  $R$  y  $E$  contiene las aristas de un árbol de expansión mínima para  $R$ .

## APLICACIÓN DE GRAFOS Y ÁRBOLES.

A partir de 1920 se despertó el interés por la teoría de gráficas. Sin duda, una de las razones de este reciente interés en la teoría de gráficas es su capacidad de aplicación en campos muy diversos, incluyendo las ciencias de la computación, la química, la investigación de operaciones, la ingeniería eléctrica, la lingüística y la economía.

### ¿Que es un grafo?

Recordemos que un grafo  $G$  es el par  $(V, A)$  que representa una relación entre un conjunto de Vértices y otro de Aristas.

Representaremos cada elemento arista como un par de elementos de  $V$ .

Gráficamente representaremos los vértices por puntos y las aristas por líneas que los unen.

Un vértice puede tener 0 o más aristas, pero toda arista debe unir exactamente 2 vértices.

Las aplicaciones más importantes de los grafos son las siguientes:

- Rutas entre ciudades.
- Determinar tiempos máximos y mínimos en un proceso.
- Flujo y control en un programa

### EJEMPLOS DE APLICACIONES DE GRÁFICAS.



**La red de metro del Distrito Federal.** Los mapas de las líneas del ferrocarril metropolitano son grafos que muestran la conectividad de las estaciones.

Los grafos son la representación natural de las redes, en las que estamos cada vez más incluidos.

Los grafos son artefactos matemáticos que permiten expresar de una forma visualmente muy sencilla y efectiva las relaciones que se dan entre elementos de muy diversa índole. Un grafo simple está formado por dos conjuntos:

- Un conjunto  $V$  de puntos llamados vértices o nodos.
- Un conjunto de pares de vértices que se llaman aristas o arcos y que indican qué nodos están relacionados.

De una manera más informal podemos decir que un grafo es un conjunto de nodos con enlaces entre ellos, denominados aristas o arcos.

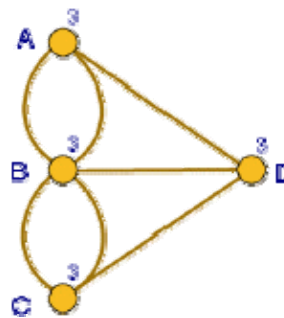
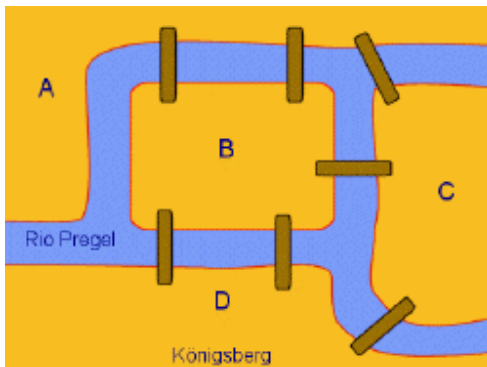
En un grafo simple entre dos nodos sólo hay un arco. Si hay más de un arco hablamos de un multígrafo. Si los arcos se pueden recorrer en una dirección concreta pero no en la contraria lo llamamos grafo dirigido o dígrafo y los arcos son entonces aristas, si los arcos salen y llegan al mismo punto formando un bucle el grafo resultante se llama pseudografo.

A pesar de que un grafo parece una estructura muy elemental, hay muchísimas propiedades de los grafos cuyo estudio ha dado lugar a una completa teoría matemática.

Fue Leonhard Euler quien ideó los grafos como una manera muy potente y elegante de resolver el problema de los puentes de Königsberg.

Königsberg (hoy Kaliningrado en Rusia) era en tiempos de Euler (siglo XVIII) una ciudad prusiana cruzada por siete puentes. Durante la época se suscitó la cuestión no resuelta de si era posible recorrer toda la ciudad cruzando cada uno de los puentes una y sólo una vez. Si hacemos una representación esquemática de la ciudad vemos que los puentes unen cuatro porciones de tierra.

La búsqueda por prueba y error no conduce a ningún resultado.



**El problema de los puentes de Königsberg. La solución de Euler.**

Esta ciudad esta recorrida por el río Pregel que crea dos islas. ¿Se puede recorrer toda la ciudad pasando una sola vez por todos y cada uno de los 7 puentes que unen la parte insular de la ciudad con el resto?

El famoso matemático abstrajo los detalles de la forma de la ciudad y sus puentes para quedarse con la conectividad, dando lugar a una de los primeros grafos. El orden de todos los vértices es impar, lo que implica que es imposible recorrerlos pasando una sola vez por cada uno.

Euler realizó una abstracción del problema representando mediante puntos las cuatro porciones de terreno y dibujando un arco entre cada dos puntos por cada puente. Llamó orden de cada vértice al numero de arcos que se reunían en el y se percató que el orden de cada vértice visitado en un recorrido sin saltos ha de ser par (sale un enlace y entra otro) excepto para dos puntos del grafo: aquellos donde se inicia y donde se acaba el recorrido, que han de tener orden impar. Si el vértice donde se inicia y se acaba son el mismo entonces todos los vértices han de ser de orden par.

En el problema de Königsberg el orden de todos los nodos es 3, esto es impar, por lo que quedó claro que no existía solución para el problema. No había un camino que recorriese todos los puentes pasando una sola vez por cada uno de ellos.

El interés de este ejemplo es que además de dar lugar a una teoría matemática muy potente los grafos se dibujan y resultan muy intuitivos, especialmente cuando los vértices son pocos. Ejemplos de grafos que todos conocemos son los organigramas que explicitan la estructura formal de la empresa, los árboles genealógicos o la circuitería de los chips electrónicos. Se usan regularmente para resolver problemas en la eficiencia del transporte, en sociología, electrónica y electricidad, detección de fraude y en general en aquellos campos en los que la conectividad es importante.

De hecho vivimos en una sociedad interconectada en la que, por definición, las redes (que son simplemente una forma de grafos dirigidos en los que cada arco tiene un valor) forman cada vez más parte de nuestra experiencia diaria. Internet es el arquetipo de la red y su conectividad nos une a todos.

Como anécdota, al parecer la captura de Saddam Hussein se realizó en parte gracias a la labor de construcción del grafo de su red de soporte, basada en las relaciones funcionales de Saddam con miembros de su partido pero sobre todo de las relaciones tribales y familiares que le unen a su ciudad natal de Tikrit.

### **Solución de problemas por medio de gráficas.**

Gracias a las gráficas podemos resolver fácilmente problemas que aparentemente son muy complicados. Resolver problemas es la principal aplicación de las gráficas.

A continuación mostraremos por medio de un ejemplo como resolver un problema por medio de las gráficas.

#### **Problema.**

¿Es posible que en un departamento de 25 personas, clasificadas según su desacuerdo, cada persona congenie con exactamente otras 5?

Para enfrentar el problema.

¿Dónde comenzar? Muchos problemas discretos se pueden resolver por medio de una grafica.

#### **Determinación de una solución.**

Un elemento fundamental al construir un modelo de gráfica es imaginar lo que esta debe ser: ¿Cuáles son los vértices y cuales las aristas? En este problema, no tenemos muchas opciones; tenemos personas y desacuerdos. Intentemos utilizar a las personas como vértices. En un modelo de gráfica, es común que las aristas indiquen una relación entre vértices. En este caso, la relación es “congeniar con”, de modo que escribiremos una arista entre los vértices (personas) si ellas congenian.

Ahora supongamos que cada persona congenia con exactamente otras cinco. Por ejemplo, en la figura que se muestra a continuación donde se muestra parte de la gráfica, Tomás congenia con Samantha, Alexandra, Juan, Bertha, y Josefina, y nadie más.





Esto implica que el grado de cada vértice es 5. Ahora la situación se resume así: Tenemos 25 vértices. En este caso y cada vértice tiene grado 5. Antes de continuar veamos si esto es posible.

El corolario dice que existe un número par de vértices de grado impar. Tenemos una contradicción, pues existe un número impar de vértices de grado impar. Por lo tanto, no es posible que en un departamento de 25 personas clasificadas según sus desacuerdos, cada persona congenie con exactamente otras cinco.

### **Solución formal.**

No es posible que en un departamento de 25 personas clasificadas según sus desacuerdos, cada persona congenie con exactamente otras 5. Supongamos por contradicción que esto es posible. Consideremos una gráfica donde los vértices sean las personas y una arista conecte 2 vértices (personas) si las personas congenian. Como cada vértice tiene grado impar, existe un número impar de vértices de grado impar, lo cual es una contradicción.

### **Resumen de técnicas para resolver problemas.**

- ❖ Muchos problemas discretos se pueden resolver modelándolos mediante gráficas.
- ❖ Para construir un modelo de gráfica, determine lo que representan los vértices y las aristas.
- ❖ En un modelo de gráfica, es muy común que las aristas indiquen una relación entre dos vértices.