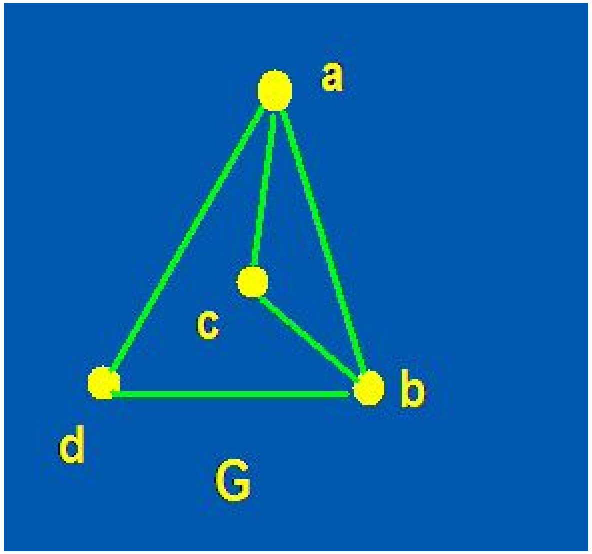
“Teoría de Grafos 1”

Dr. Said Polanco Martagón

Tipos de Grafos

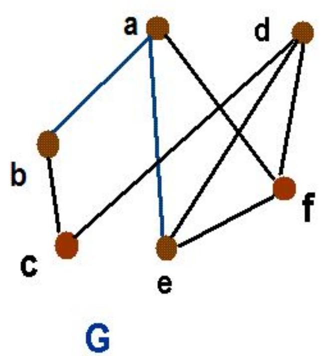
- Un grafo **G** es un par **(V,E)** donde:
  - $\mathbf{V = \{v_1,\ldots,v_n\}}$ es un conjunto de vértices
  - $\mathbf{E = \{e_1,\ldots,e_m\}}$ es un conjunto de aristas,

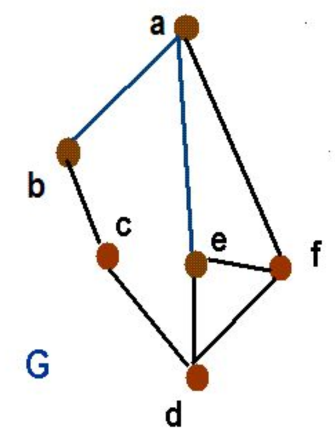
    con cada $\mathbf{e_k} \in \mathbf{\{v_i, v_j\}}$, con $\mathbf{v_i, v_j} \in \mathbf{V}$, $\mathbf{v_i \neq v_j}$
- Los vértices se representan como puntos y las aristas como líneas entre vértices
- Ejemplo:
  - **G = (V,E)**
  - **V = {a,b,c,d }**
  - **E = {{a,b}, {b,c}, {a,c}, {a,d}, {d,b} }**



Tipos de grafos

- Es importante recordar que un mismo grafo puede tener diferentes representaciones gráficas
- **Ejemplo**:



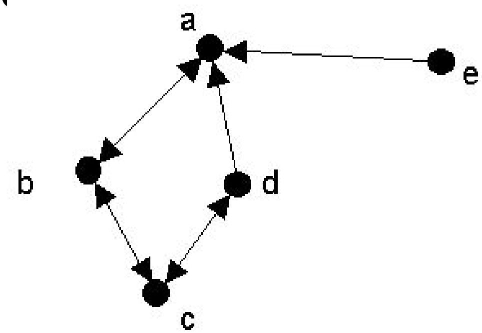


Dos representaciones del mismo grafo

G = ({a,b,c,d,e,f},{{a,b},{a,e},{a,f}{e,f},{b,c},{c,d},{e,d},{d,f}})

Tipos de Grafos

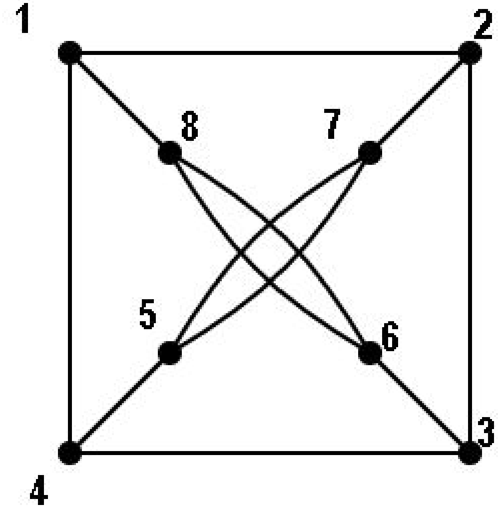
- Si el orden influye en la aristas se habla de **grafos dirigidos**:



- En este caso a las aristas se les llama **arcos** y se representan como pares para indicar el orden:
  - **V = { a,b,c,d,e}**
  - **A ={(e,a), (a,b), (b,a), (d,a), (c,d), (d,c),(b,c),(c,b) }**

Tipos de Grafos

- Si se permite que haya más de una arista se habla de **multigrafos**:



Tipos de Grafos

- Cuando las aristas tienen un valor numérico asociado se llama de **grafos valorados**:

- Al valor numérico asociado se le llama **coste** de la arista

Tipos de Grafos

- Los tipos anteriores pueden combinarse, dando lugar por ejemplo a **multigrafos valorados**, o **grafos dirigidos valorados**, etc.

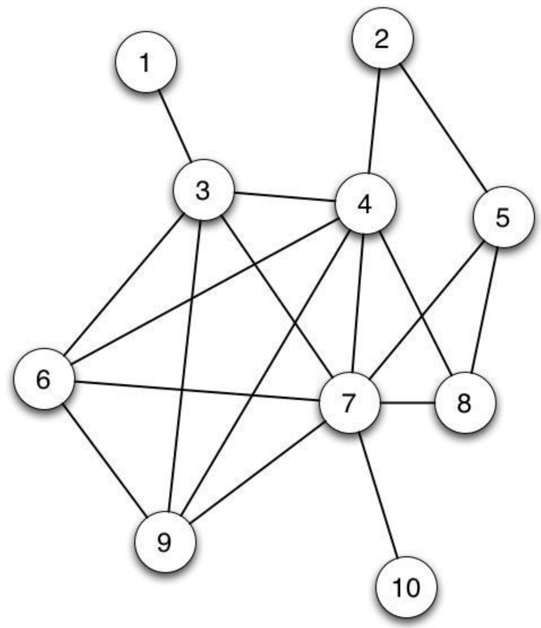
- En el resto del tema cuando no se diga lo contrario G representará un **grafo o multigrafo no dirigido**

Conceptos Básicos

- Dos vértices se dicen **adyacentes** si existe una arista que los une
- Los vértices que forman una arista son los **extremos** de la arista
- Si **v** es un extremo de una arista **a**, se dice que **a** es **incidente** con **v**
- El grado de un vértice **v**, **gr(v)** es el número de aristas incidentes en **v.** Si hace falta indicar el grafo en el que está **v** escribiremos **gr(G,v)**

Conceptos Básicos

- **Ejemplo**:



- gr(6)= ________ gr(1) = __________

Conceptos Básicos

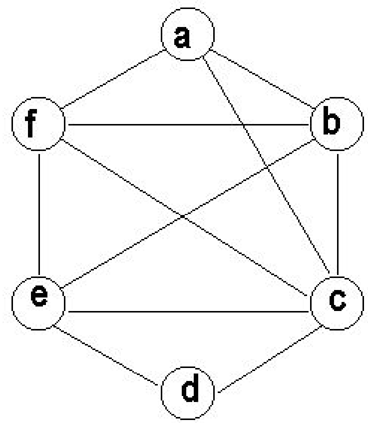
- **Teorema** (de los “apretones de manos”)

Sea G=(V,A) un grafo. Entonces: $\sum_{v \in V} \mathbf{gr(v)} = \mathbf{2|A|}$

- Significado: la suma de los grados de todos los vértices es igual a 2 veces el número de aristas
- **Explicación**:

Conceptos Básicos

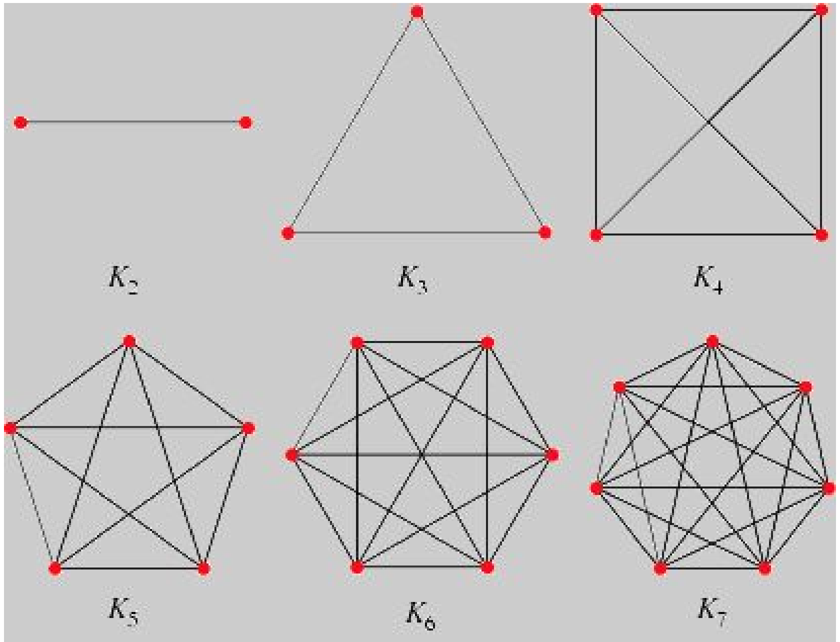
- **Ejemplo**:



- gr(a)+gr(b)+gr(c)+gr(d)+gr(e)+gr(f) = 3+4+5+2+4+4 = **22**
- 2|A| = 2 _____ = ______

Conceptos Básicos

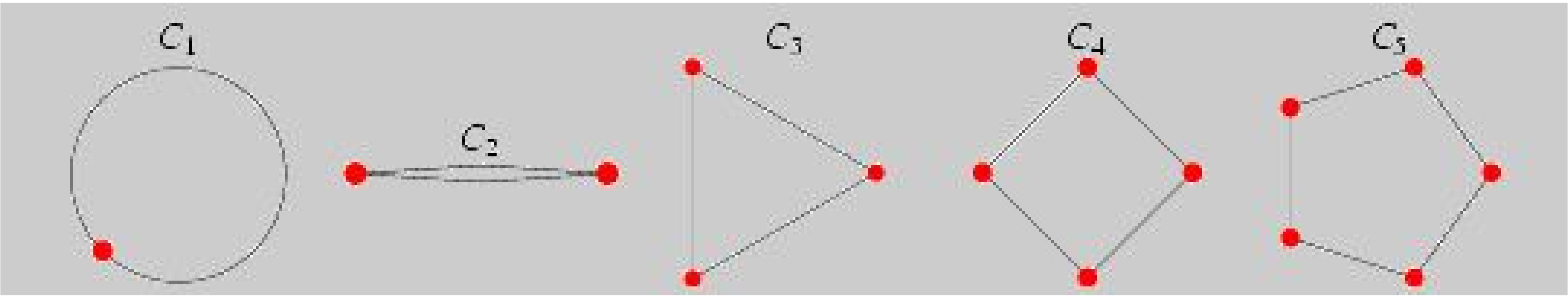
- Para cada **n≥1** se llama **grafo completo** de orden n, y se representa por Kn, al grafo de n vértices conectados de todas las formas posibles:



- **Pregunta**: ¿Cuántas aristas tiene en general Kn?

Conceptos Básicos

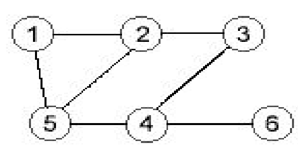
- Se llama **ciclo** de grado n, y se denota **Cn**, a $\mathbf{G=(\{v_1,\ldots,v_n\},}$
  $$\mathbf{\{\{v_1, v_2\}, \{v_2, v_3\},\ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}\}\ )}$$



- **Nota**: A menudo sólo se consideran ciclos para $n \geq 3$

Representación de Grafos

- Para representar los grafos a menudo se utiliza la llamada **matriz de adyacencia**
- Se construye imaginando que en las filas y las columnas corresponden a los vértices. Se pone un 0 para indicar que 2 vértices no son adyacentes, y un 1 para indicar que sí lo son:



G

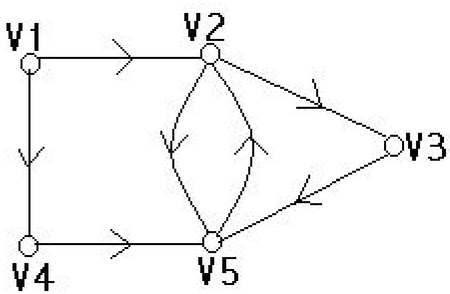
$$
\begin{array}{c c}
 & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}
\end{array}
$$

Matriz de Adyacencia de G

- Para representarla en un ordenador se utilizan matriz de valores lógicos (*booleanos*). True → hay arista, False→ no hay arista

Representación de Grafos

- En el caso de un grafo no dirigido la matriz será simétrica. No ocurre lo mismo para grafos dirigidos:

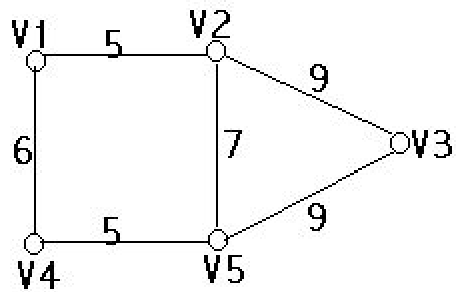


$$
\begin{array}{c|c}
 & \begin{array}{ccccc} V1 & V2 & V3 & V4 & V5 \end{array} \\
\hline
\begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

- Se supone que la **fila** representa el vértice **origen**, y la **columna** el vértice **destino** del arco

Representación de Grafos

- La matriz de adyacencia también permite representar **grafos valorados**

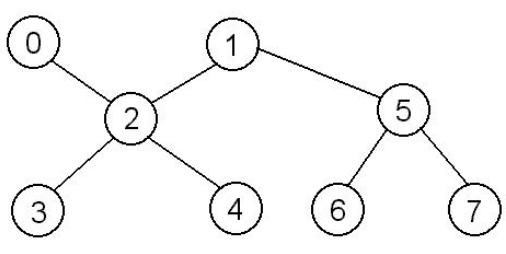


$$
\begin{array}{c|c}
 & \begin{array}{ccccc} V1 & V2 & V3 & V4 & V5 \end{array} \\
\hline
\begin{array}{c} V1 \\ V2 \\ V3 \\ V4 \\ V5 \end{array} &
\begin{pmatrix}
0 & 5 & 0 & 6 & 0 \\
5 & 0 & 9 & 0 & 7 \\
0 & 9 & 0 & 0 & 9 \\
6 & 0 & 0 & 0 & 5 \\
0 & 7 & 9 & 5 & 0
\end{pmatrix}
\end{array}
$$

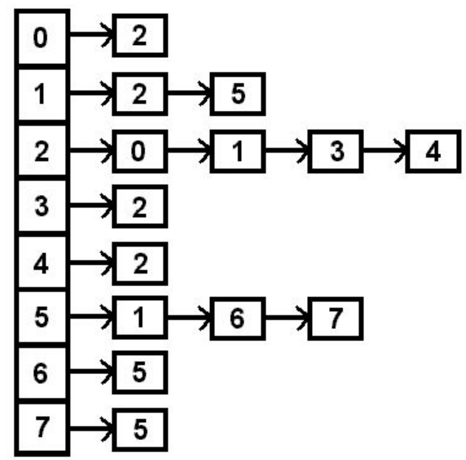
- El valor guardado es el **coste** de la arista/arco
- En lugar de **0**, a menudo se emplea un valor especial **∞** para indicar que dos vértices no están conectados

Representación de Grafos

- En informática a menudo en lugar de la matriz se usa la **lista de adyacencia**
- A cada vértice le corresponde una lista con sus adyacentes:



G



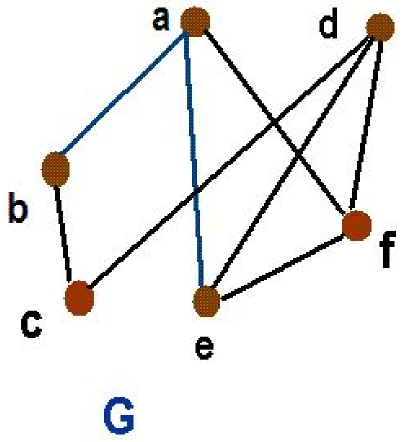
Lista de Adyacencia de G

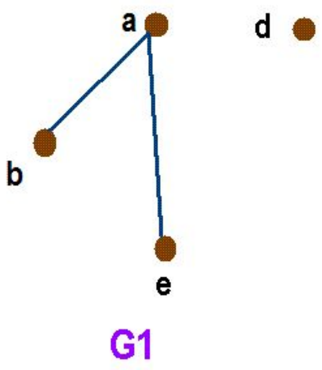
Subgrafos

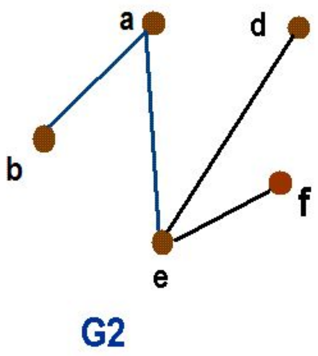
- Sea **G=(V,A)**. **G'=(V',A')** se dice **subgrafo** de **G** si:
  1. **V'** $\subseteq$ **V**
  2. **A'** $\subseteq$ **A**
  3. **(V',A')** es un grafo
- Resultado fácil de comprobar:
  - Si **G'=(V',A')** es subgrafo de G, para todo **v** $\in$ **G** se cumple **gr(G',v)≤ gr(G,v)**

Subgrafos

- **Ejemplo**:



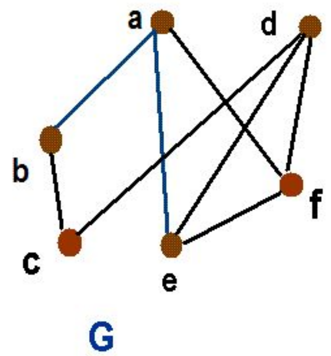




- G1 y G2 son subgrafos de G

Subgrafos

- Un grafo se dice cíclico cuando contiene algún ciclo como subgrafo
- Ejemplo:



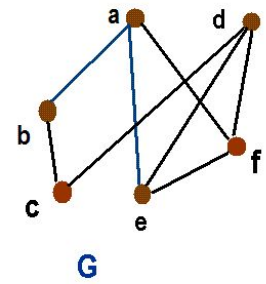
- Contiene dos ciclos de long. 3: {a,e,f,a} y {_, _, _, _}
- Contiene un ciclo de longitud 6: {_,_,_,_,_,_,_}
- ¿Contiene algún ciclo más? ___

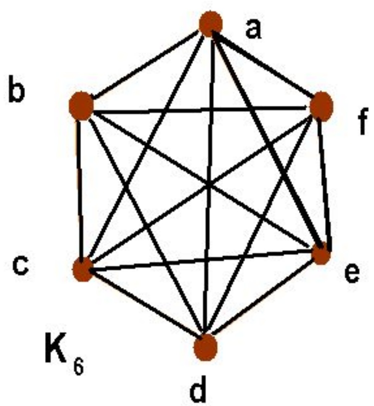
Grafo Complementario

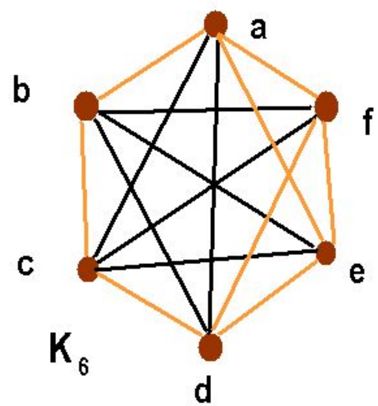
- El complementario **G'** de un grafo **G=(V,A)** tiene:
  - Los mismos vértices que **G**
  - Si **{u,v}** $\in$ **G**, entonces **{u,v}** $\notin$ **G'**
  - Si **{u,v}** $\notin$ **G**, entonces **{u,v}** $\in$ **G'**
- Una forma de construirlo:
  - Dibujar el corresp. grafo completo **Kn**, con **n=|V|**
  - Eliminar de **Kn** las aristas **{u,v}** $\in$ **G**

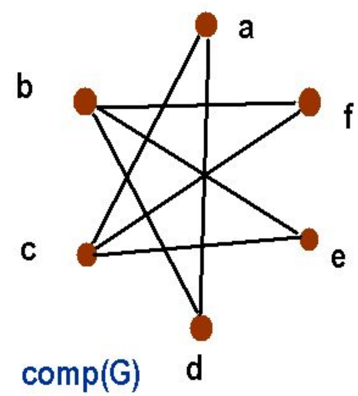
Grafo complementario

- **Ejemplo** : Complementario de









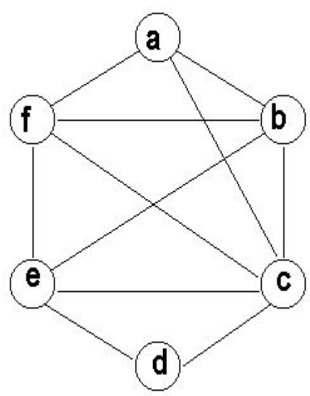
1º Representar K6

2º Marcar las aristas de G

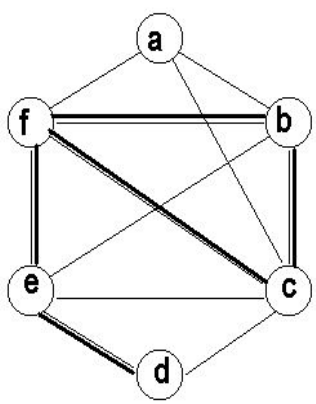
3º Eliminarlas

Caminos y conectividad

- Un **recorrido** en un grafo **G = (V,A)** es una sucesión de vértices $\mathbf{v_0, v_1, \ldots, v_k}$ tal que $\{\mathbf{v_i, v_{i+1}}\} \in \mathbf{A}$ para todo $\mathbf{0 \leq i < k}$
- La **longitud** de un recorrido $\mathbf{v_0, v_1, \ldots, v_k}$ es **k**
- **Ejemplo**:



G



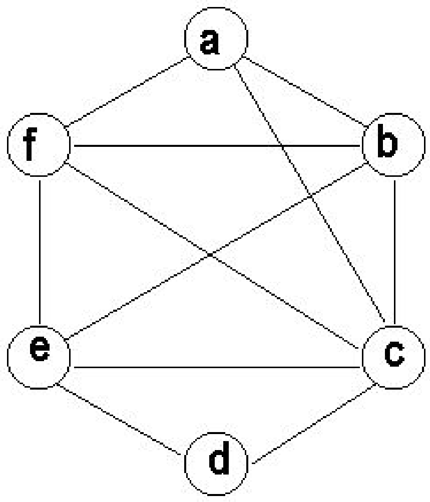
f,b,c,f,e,d es un recorrido de longitud 5 sobre G

Caminos y conectividad

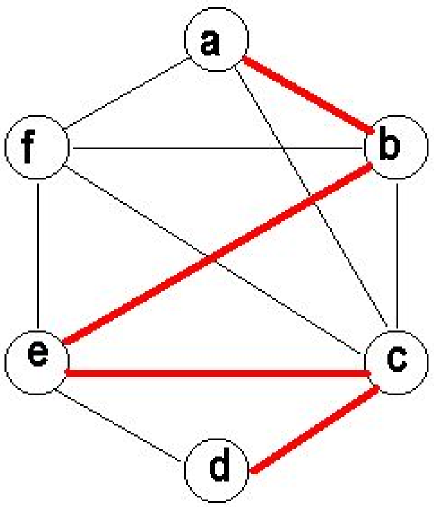
- **Observación**: Un recorrido puede repetir vértices, y puede comenzar y acabar en vértices diferentes
- Un **camino** es un recorrido $\mathbf{v_0, v_1, \ldots, v_k}$ en el que $\mathbf{v_i \neq v_j}$ para $\mathbf{0 \leq i,j \leq k,}$ con $\mathbf{i \neq 0}$ o $\mathbf{j \neq k}$
- Es decir en un camino todos los vértices son **distintos** entre sí, excepto quizás el primero y el último

Caminos y conectividad

- Ejemplo:



G



a,b,e,c,d es un camino

Caminos y conectividad

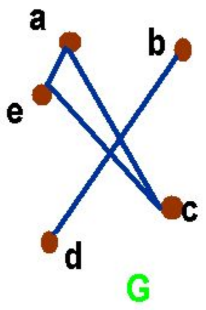
- Si existe un camino entre dos vértices se dice que están **conectados**
- Sea **G=(V,A)** un grafo. La relación

  xRy $\leftrightarrow$ x e y están conectados

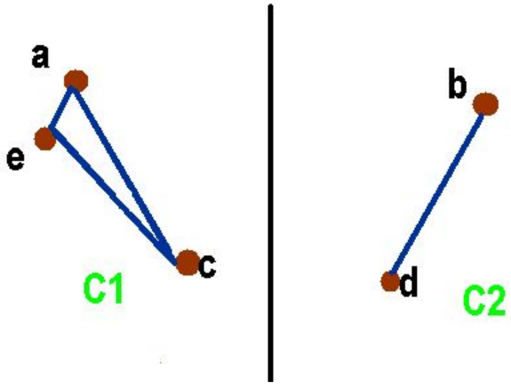
  es de equivalencia (R $\subseteq$ ___)
- Si para todo par de vértices de un grafo están conectados se dice que el grafo es **conexo** g
- Las **componentes conexas** de un grafo G son los mayores subgrafos conexos de G

Caminos y conectividad

- **Ejemplo**. Consideramos el grafo:

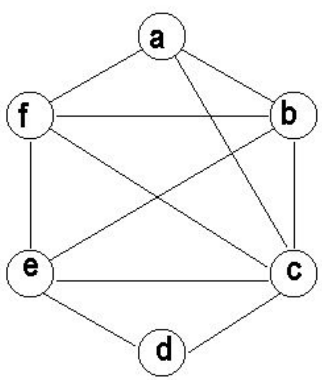


- Se tiene que:
  - G no es conexo: no hay camino entre a y b, por ejemplo.
  - [a] = {a,c,e} [c] = {a,c,e} [e]={a,c,e} [b]={b,d} [d]={b,d}
  - G/R = {[a],[b]}
  - G tiene dos componentes conexas

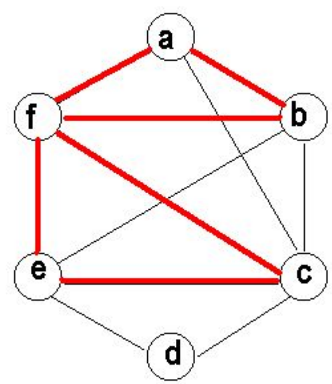


Caminos y conectividad

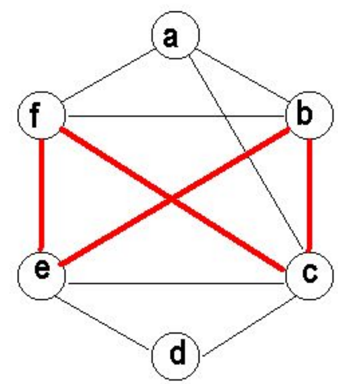
- Un recorrido $\mathbf{v_0, v_1, \ldots, v_k}$ tal que $\mathbf{v_0 = v_k}$ es un **circuito**
- Un camino $\mathbf{v_0, v_1, \ldots, v_k}$ tal que $\mathbf{v_0 = v_k}$ es un **ciclo**



G



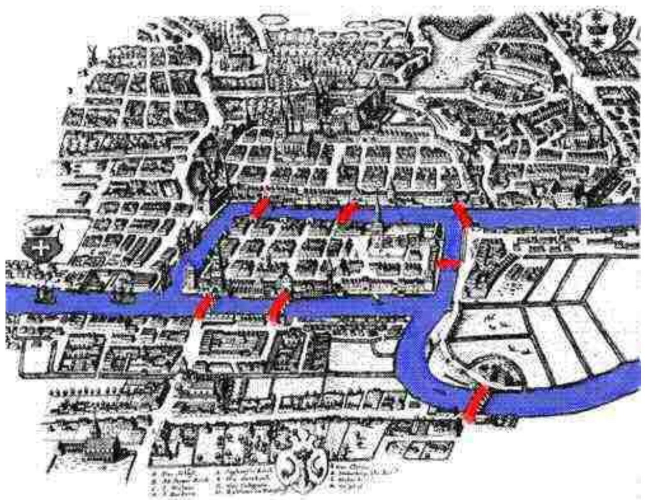
a,b,f,c,e,f,a es un circuito



f,c,b,e,f es un ciclo

Recorridos eulerianos

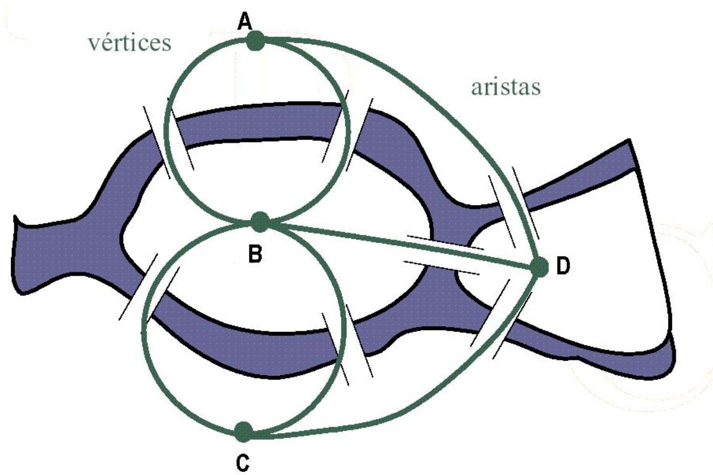
- Ciudad de Könisberg, en XVIII:

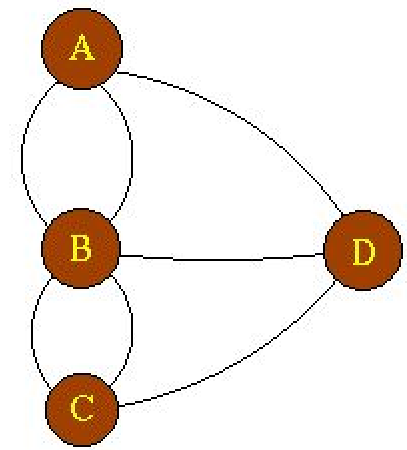


- **Pregunta**: ¿sería posible dar un paseo pasando por cada uno de los siete puentes, sin repetir ninguno, comenzando y acabando en el mismo punto?

Recorridos eulerianos

- Representación propuesta por Leonard Euler en 1736:





- ¿Existe un circuito que pase por todas las aristas una sola vez?

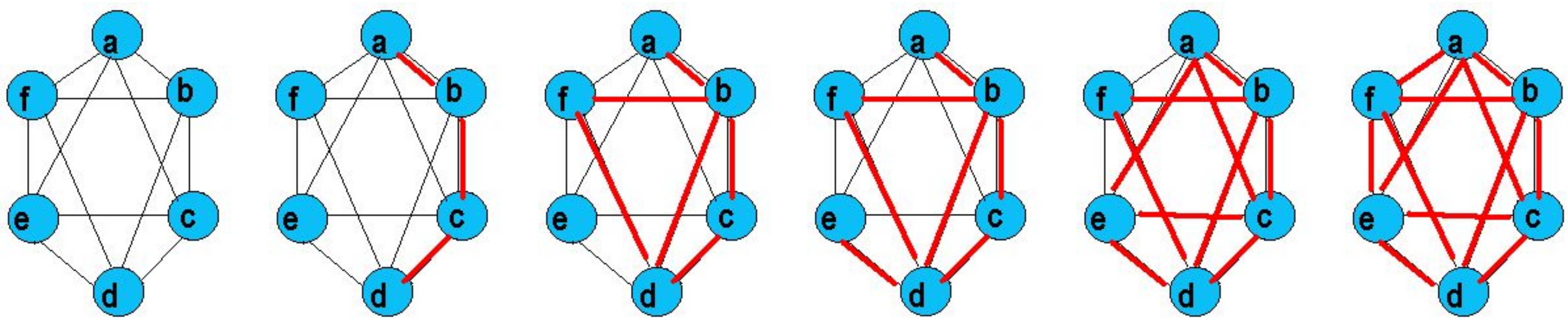
Recorridos eulerianos

- A estos circuitos se les llama **circuitos eulerianos**, y a los grafos que los contienen **grafos eulerianos**
- **Grafo o multigrafo euleriano**: admite un recorrido que pasa por todas las aristas una sola vez, empezando y terminando en el mismo vértice. Los vértices sí se pueden repetir
- **Ejemplo**: Grafo euleriano.

Circuito euleariano: **a,b,c,d,b,f,d,e,a,c,e,f,a**

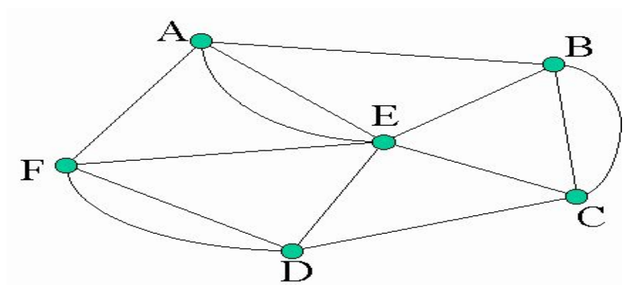
Recorridos eulerianos

- **Ejemplo**: Grafo euleriano.



Circuito euleariano: **a,b,c,d,b,f,d,e,a,c,e,f,a**

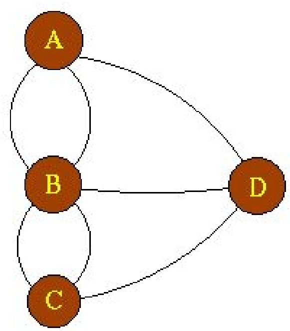
- **Ejemplo**: El siguiente grafo es euleriano



Encuentra un circuito euleriano:

Recorridos eulerianos

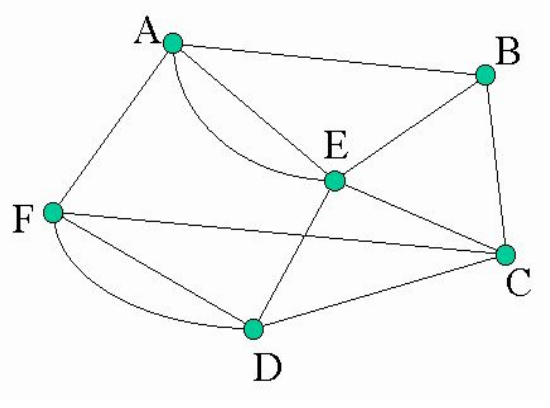
- ¿Cómo saber si un grafo (o multigrafo) es euleriano?
- **Teorema de Euler**: Un grafo conexo es euleriano ←→ no tiene vértices de grado impar
- **Ejemplo**:

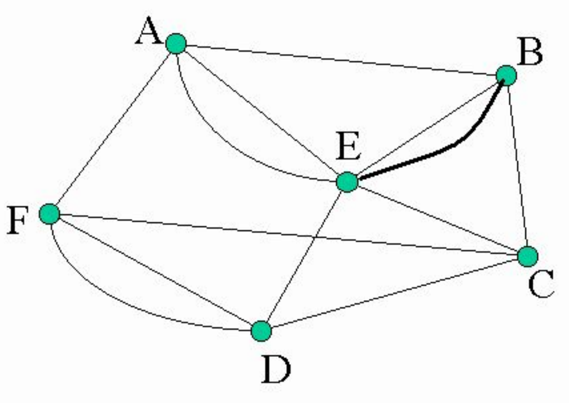


A tiene grado 3→ el grafo de los puentes no es euleriano.

Recorridos eulerianos

- Si el grafo/multigrafo tiene sólo dos vértices de grado impar se llama **semi-euleriano**. Se puede convertir en euleriano añadiéndole una arista:





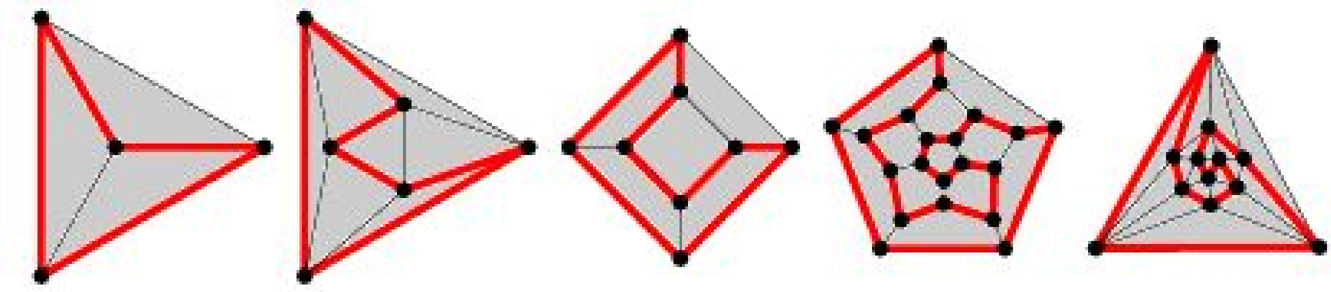
Semi-euleriano

(__,__ grado impar)

Euleriano

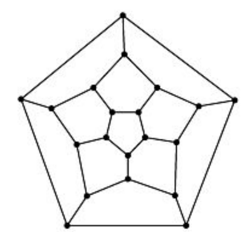
Recorridos hamiltonianos

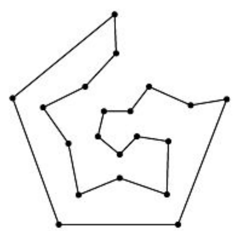
- Un grafo se dice **hamiltoniano** si existe un ciclo que recorre todos sus vértices. Al ciclo se le llama **ciclo hamiltoniano**
- **Ejemplos:**

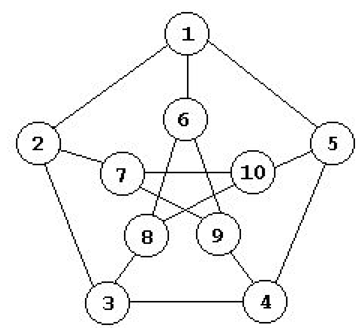


Recorridos hamiltonianos

- **No existe** un método sencillo para saber si un grafo es no hamiltoniano → problema muy complejo
- **Ejemplo**: Este grafo es hamiltoniano

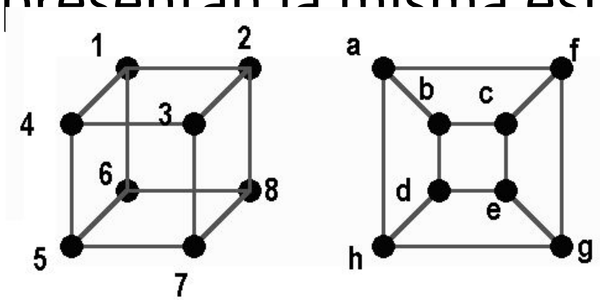


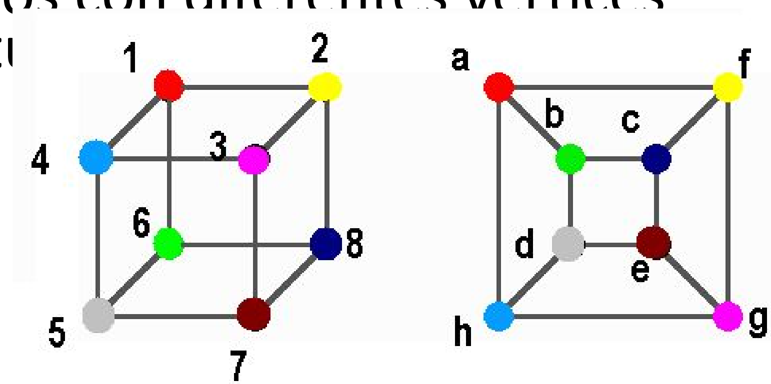
- ...pero este no ¡difíc  obar!



Isomorfismo de grafos

- **Idea**: En ocasiones dos grafos con diferentes vértices presentan la misma estructu



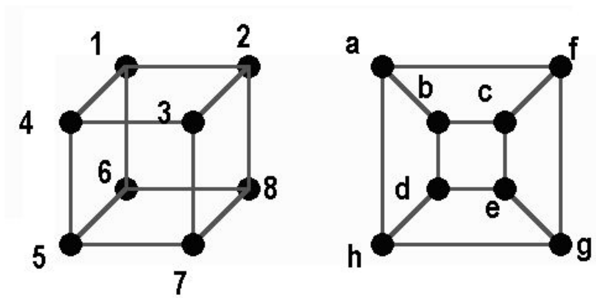


¿Cómo probarlo? Buscando una función biyectiva que convierta los vértices de uno en otro, preservando la estructura de las aristas

- **Definición**: Dos grafos **G=(V,A)**, **G’=(V’,A’)** son **isomorfos** si existe una función biyectiva **f:V→V’** tal que **{a,b}∈A ↔ {f(a),f(b)}∈A’**

Isomorfismo de grafos

- **Ejemplo:**



$f(1) = \mathbf{a}$ $f(2) = \mathbf{f}$ $f(6) = \mathbf{b}$
$f(4) = \mathbf{h}$ $f(5) = \mathbf{d}$ $f(3) = \mathbf{g}$
$f(7) = \mathbf{e}$ $f(8) = \mathbf{c}$

Los dos grafos son isomorfos. **Demostración**: Construimos f como se indica al lado de la figura. Se tiene que:

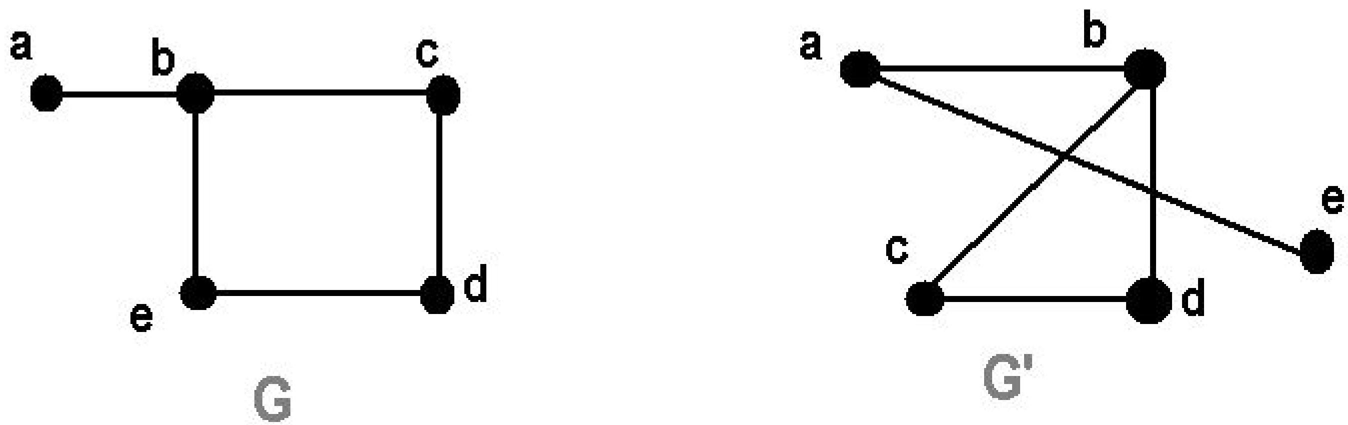
{1,2}→f→{a,f} {6,8}→f→{b,c} {1,6}→f→{a,b} {2,8}→f→{f,c} {4,3}→f→{h,g} {1,4}→f→{a,h} {2,3}→f→{f,g} {5,7}→f→{d,e} {4,5}→f→{h,d} {3,7}→f→{g,e} {6,5}→f→{b,d} {8,7}→f→{c,e}

Isomorfismo de grafos

- ¿Y como saber si dos grafos **no son isomorfos**?
- Hay que buscar alguna característica que diferencie la estructura de los dos grafos, como por ejemplo:
  - Distinto número de vértices o de aristas
  - Distinto número de ciclos de una longitud dada
  - Distinto número de vértices con un mismo grado n
  - Aristas conectando vértices con dos grados tales que no existan aristas de las mismas características en el otro grafo

Isomorfismo de grafos

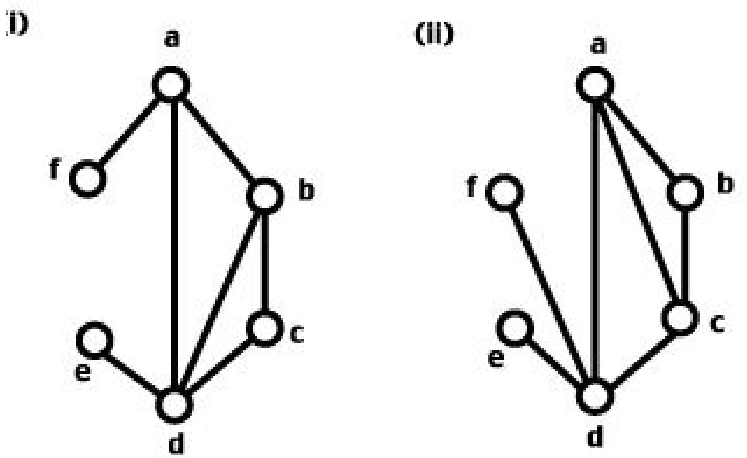
- **Ejemplo**: ¿son isomorfos estos dos grafos?



- **Respuesta**: no; G’ tiene un ciclo de longitud 3 (b,d,c,b) y G no tiene ninguno de longitud 3

Isomorfismo de grafos

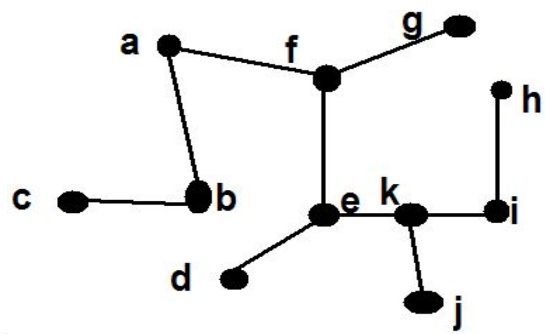
- ¿Son isomorfos? ___



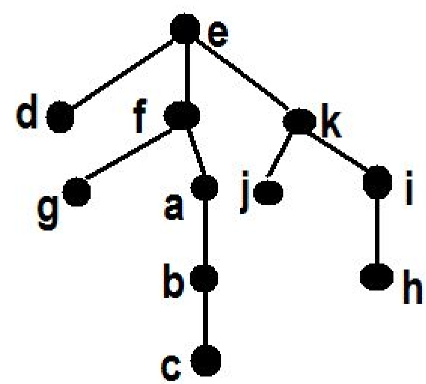
- ¿por qué? ________________________________-

Árboles

- **Árbol**: Grafo conexo y sin ciclos
- **Ejemplo**:

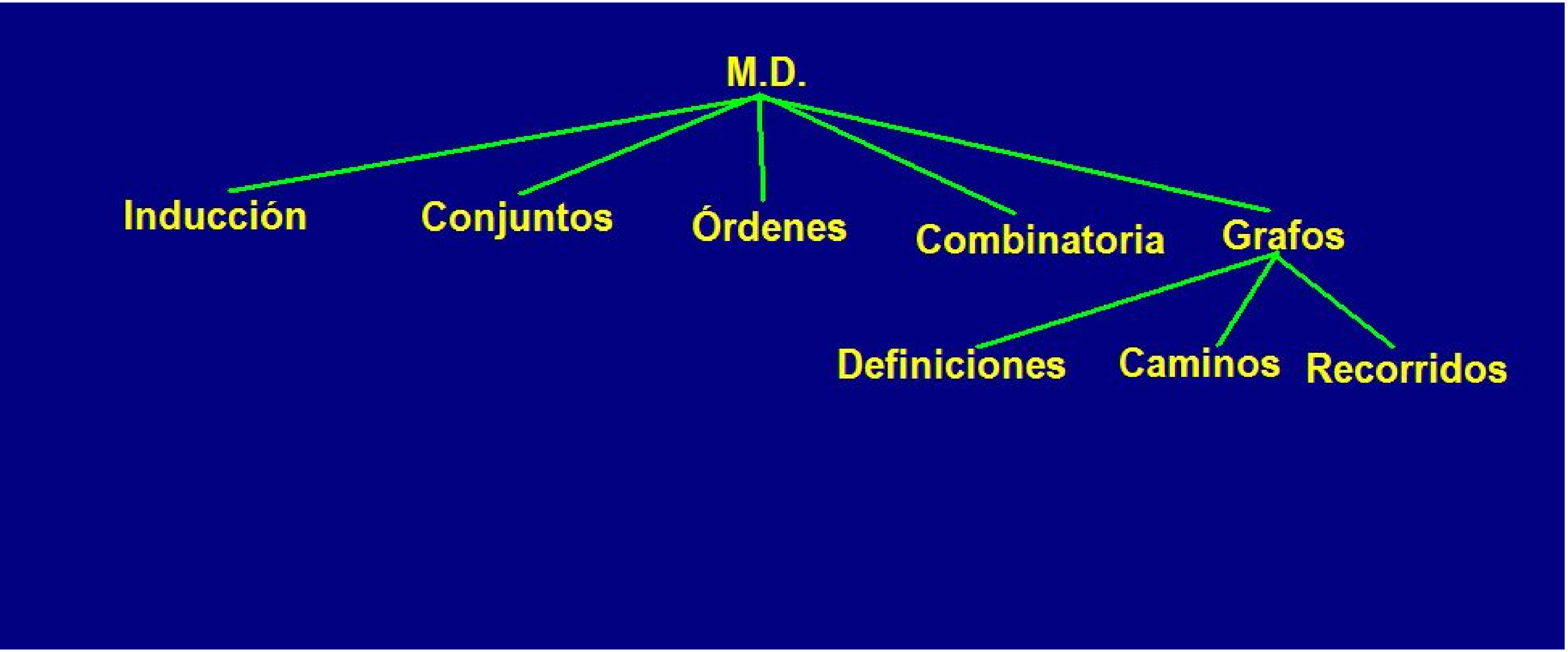


- A menudo se selecciona un nodo especial al que se llama **raíz**, y se dibuja con la raíz en la parte superior, sus adyacentes más abajo y así sucesivamente:



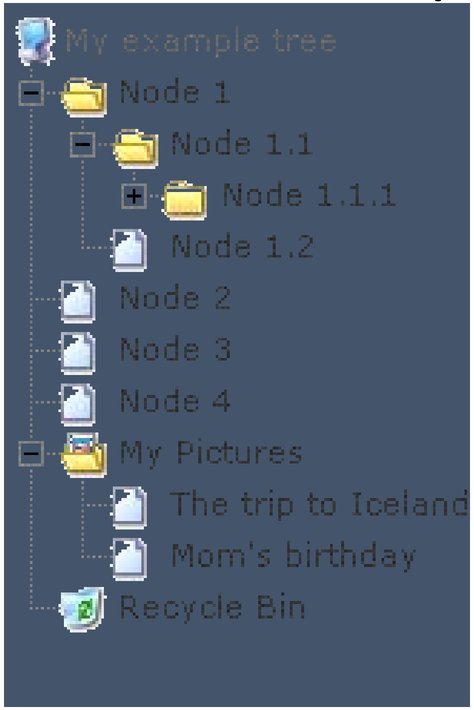
Árboles

- Ejemplo: árbol



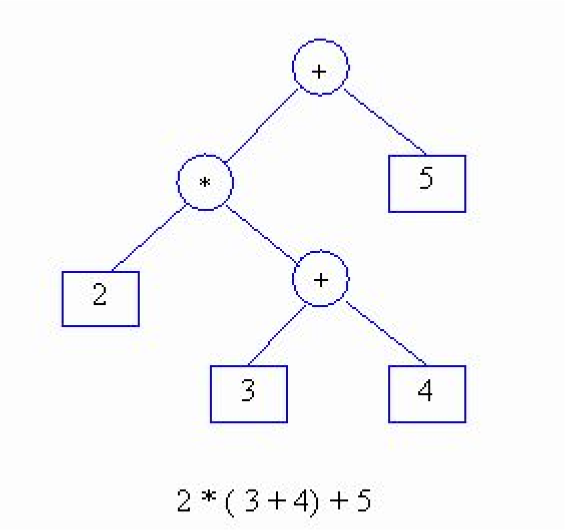
Árboles

- **Ejemplo**: Una estructura de carpetas y ficheros es un árbol

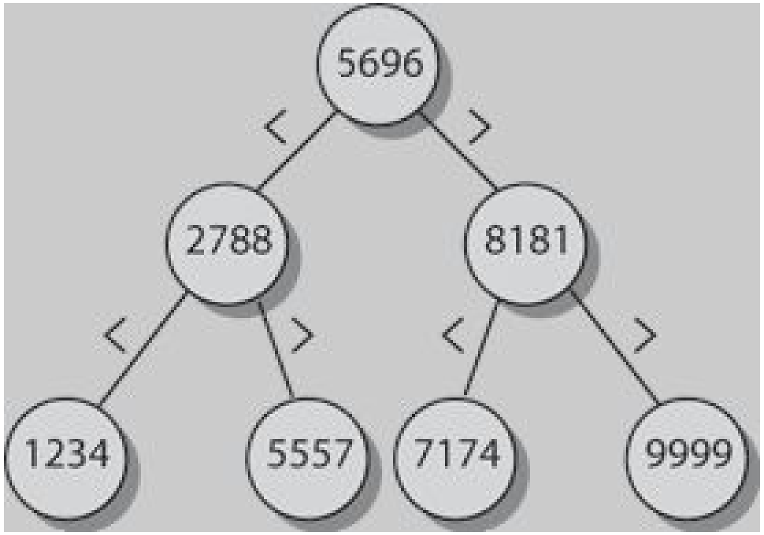


Árboles

- **Ejemplos**:



Análisis de expresiones



Árboles de búsqueda

Árboles

- Un poco de **terminología**
  - Los vértices de un árbol se llaman **nodos**
  - Los nodos descendientes inmediatos de un nodo son sus **hijos**, y el nodo superior es el **padre**
  - A una secuencia descendente de nodos se le llama **rama**
  - Los nodos sin hijos se llaman **hojas**, y los que sí tienen hijos **nodos internos**
  - Un conjunto de árboles es un **bosque**

Árboles

- Algunas **propiedades**.

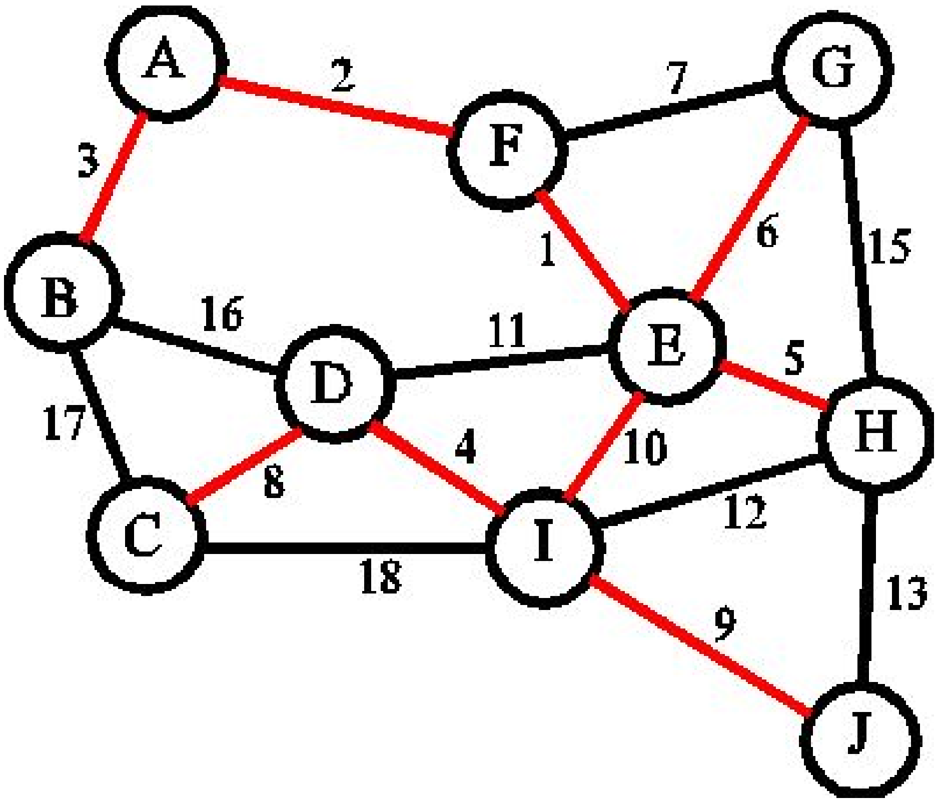
  Sea **G =(V,A)** un árbol. Entonces:

  - Entre cada par de vértices x,y hay un único camino
  - Al quitar de A cualquier arista resulta un bosque con 2 árboles
  - Al añadir una arista nueva siempre se obtiene un ciclo
  - $|A| = |V| -1$

Árboles recubridores

- Dado un grafo conexo **G =(V,A)** decimos que un árbol T **=(V',A')** es un árbol recubridor de **G** si **V=V'**, y **A** $\subseteq$ **A'**.
- En el caso de grafos valorados interesa que la suma de pesos de las aristas del árbol sea lo más pequeña posible: árbol de recubrimiento mínimo.

Árbol de recubrimiento mínimo



Algoritmo de Prim

- Se usa para construir árboles recubridores:
  1. Se elige un vértice cualquiera del grafo como vértice inicial y se marca.
  2. Mientras que queden vértices no marcados elegimos un vértice no marcado que esté conectado con alguno marcado. Marcamos tanto el vértice como una de las aristas que lo unen con los ya marcados
- En el caso de grafos valorados en cada paso se toma la arista de menor peso que cumpla 2) y se obtiene un árbol de recubrimiento mínimo.

Algorithm 7.4 PRIM

Input: A weighted connected undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$.

Output: The set of edges T of a minimum cost spanning tree for G.

```
T ← {};    X ← {1};    Y ← V − {1}
for y ← 2 to n
    if y adjacent to 1 then
        N[y] ← 1
        C[y] ← c[1, y]
    else C[y] ← ∞
    end if
end for
for j ← 1 to n − 1        {find n − 1 edges}
    Let y ∈ Y be such that C[y] is minimum
    T ← T ∪ {(y, N[y])}       {add edge (y, N[y]) to T}
    X ← X ∪ {y}               {add vertex y to X}
    Y ← Y − {y}               {delete vertex y from Y}
    for each vertex w ∈ Y that is adjacent to y
        if c[y, w] < C[w] then
            N[w] ← y
            C[w] ← c[y, w]
        end if
    end for
end for
```