



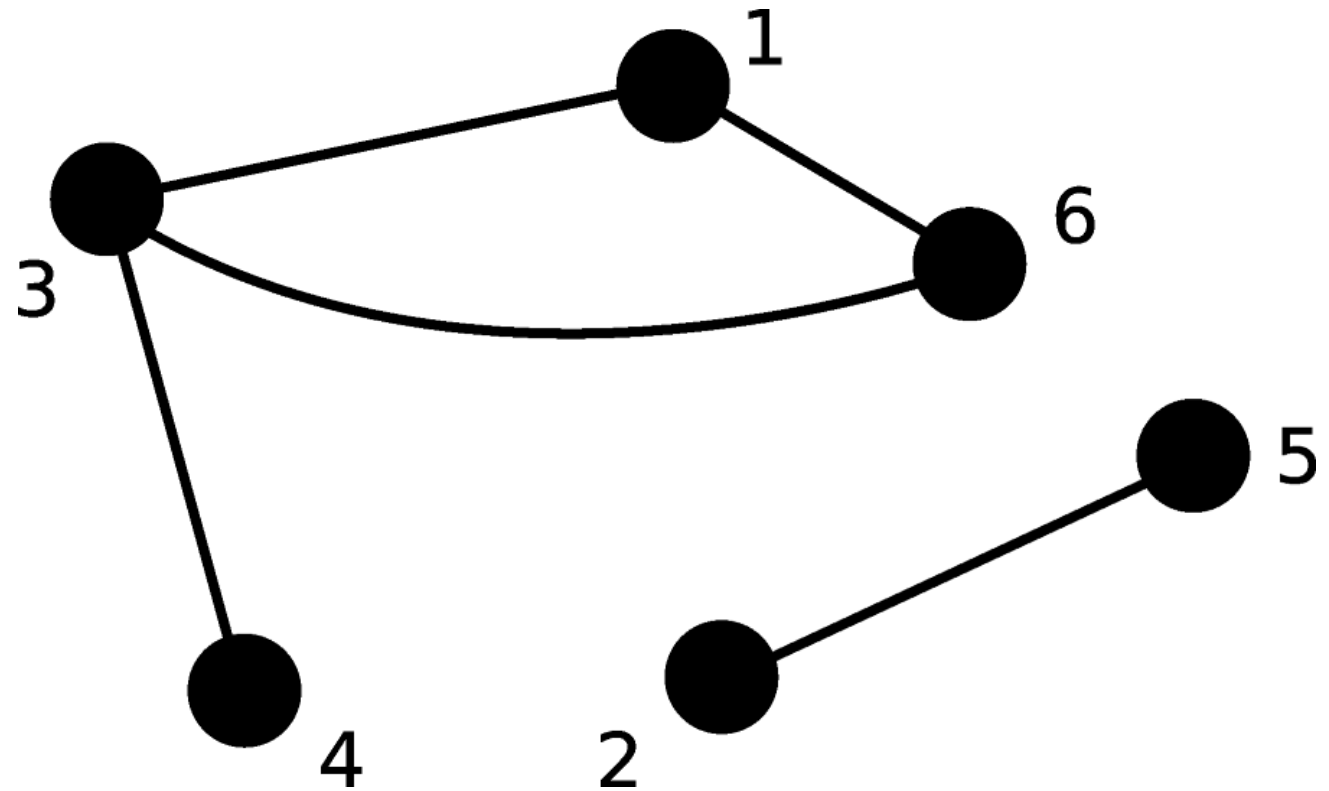
Teoría de Grafos

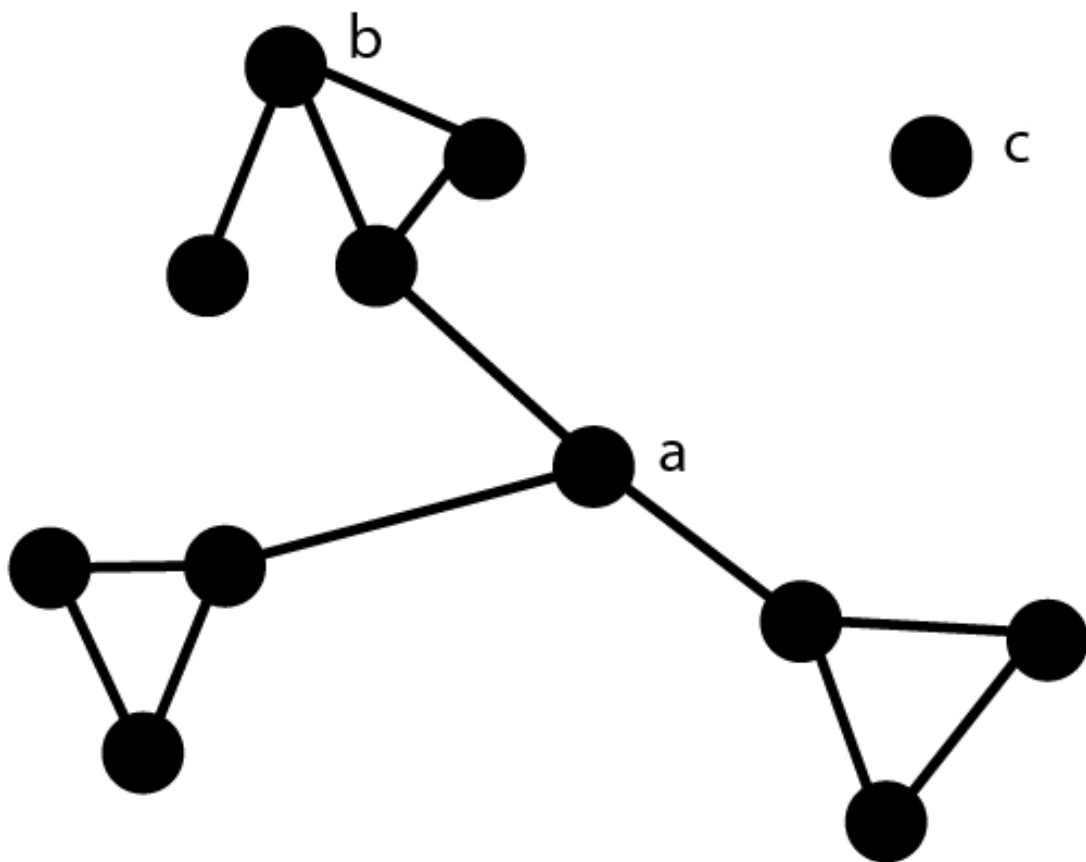
¿Qué es la teoría de grafos?

La teoría de grafos es la rama de las matemáticas computacionales que estudia las relaciones en los grafos, los cuales están formados por nodos y vértices



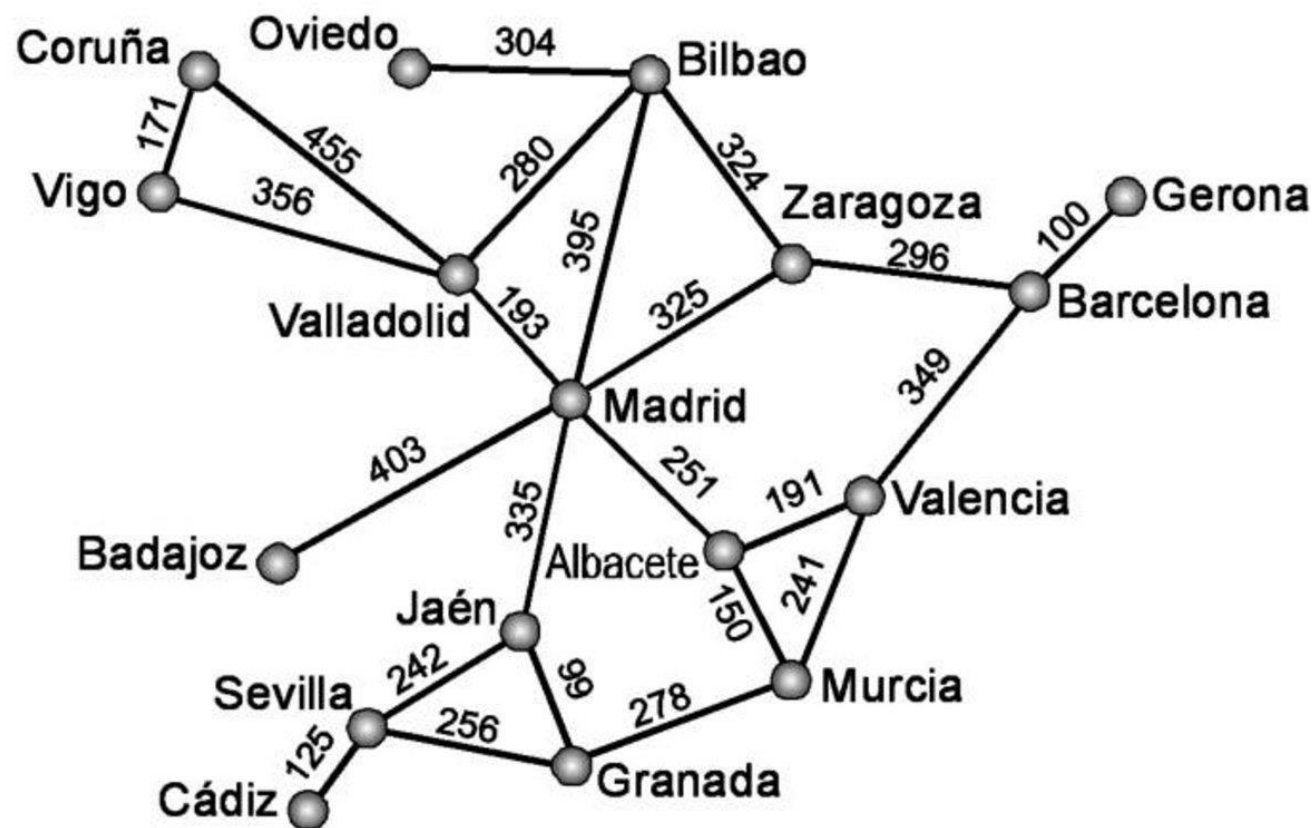
Ejemplo de grafo 1





**Ejemplo de
grafo 2**

Ejemplo de grafo 3

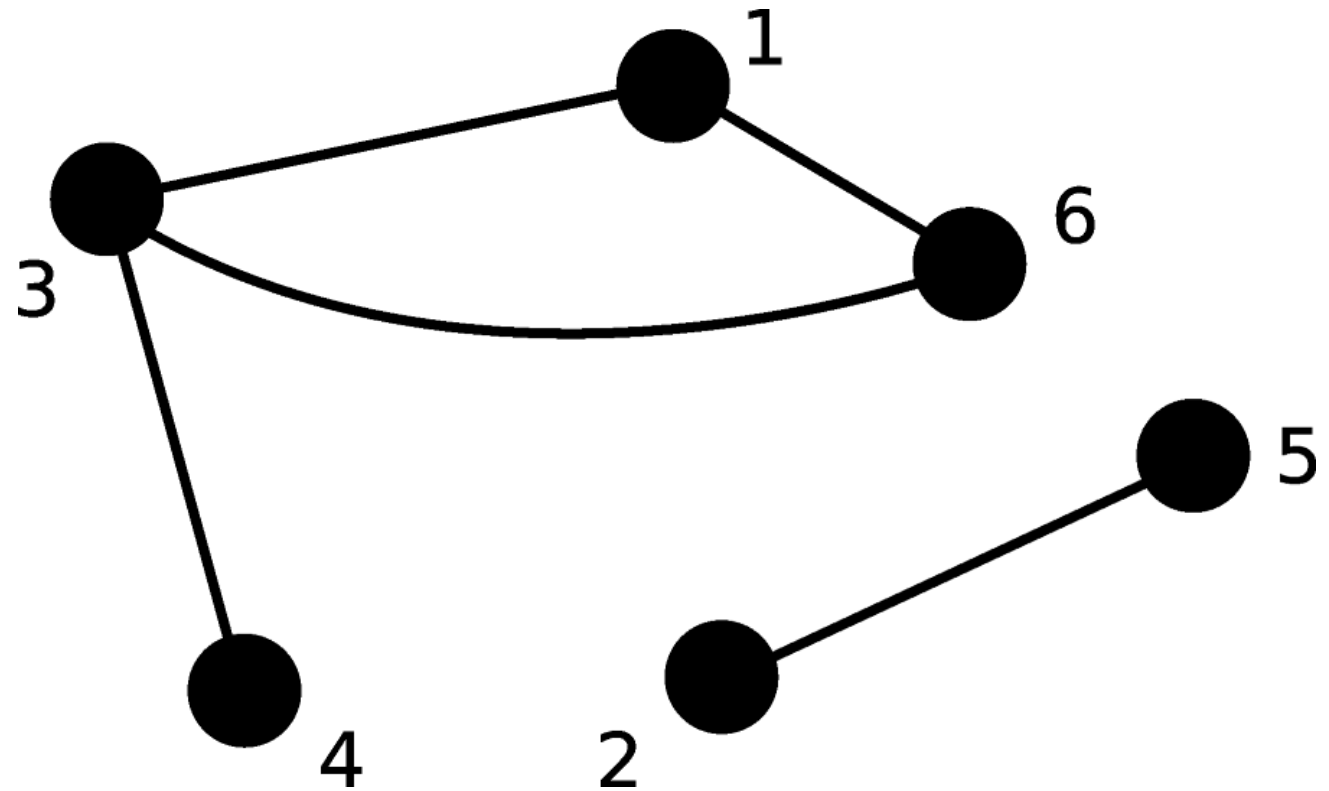


Conectando nodos en los grafos

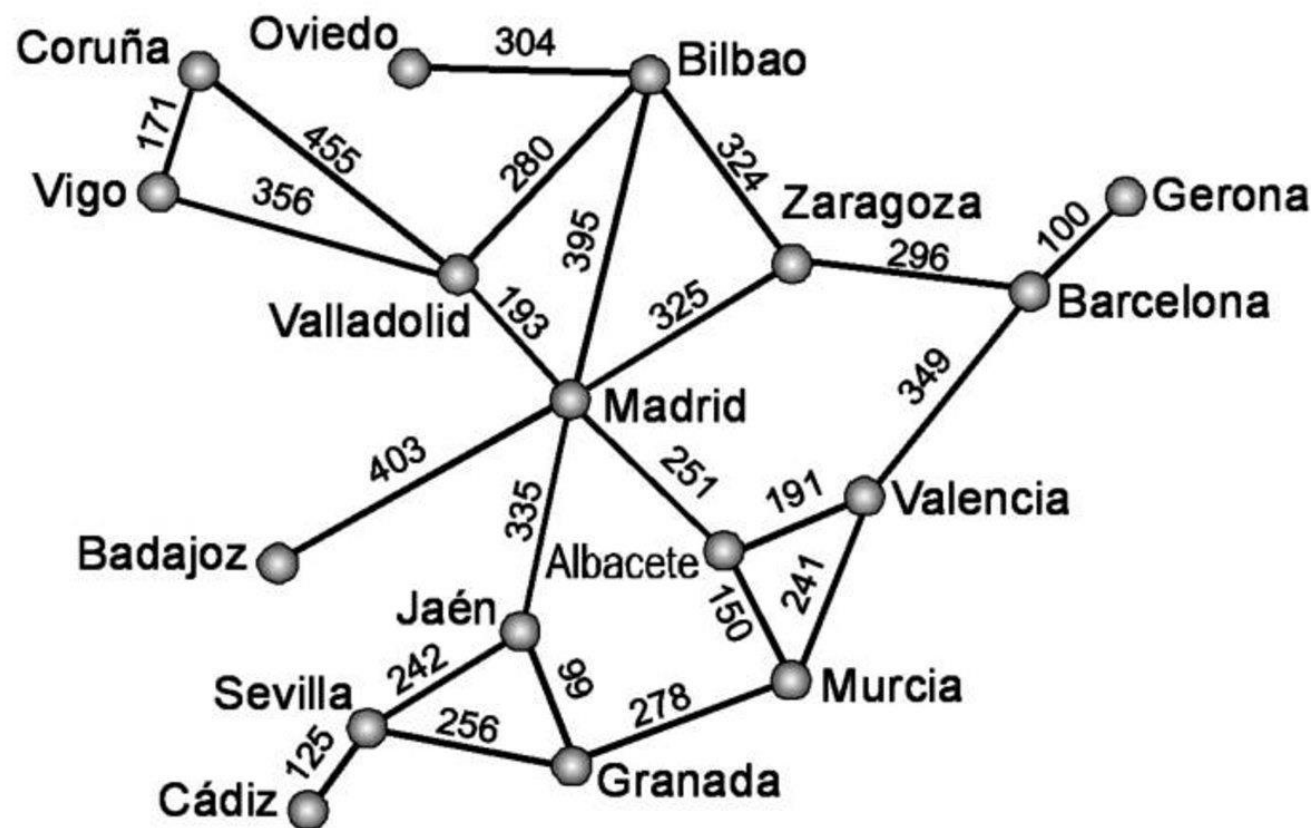
Dos nodos se pueden conectar de distintas formas, así como se puede llegar al mismo nodo pasando por las distintas aristas del grafo

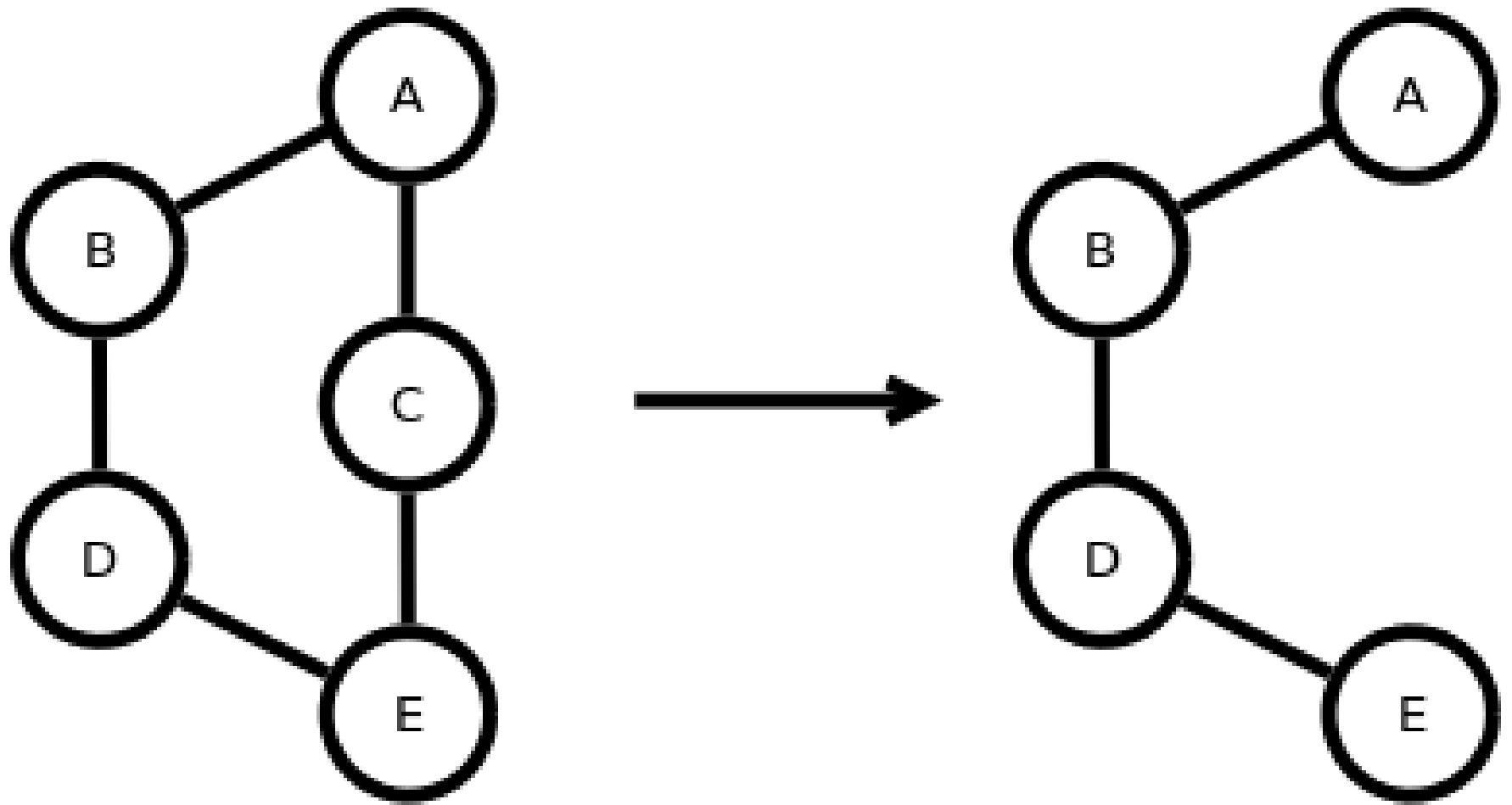
Vértice(s) repetido(s)	Arista(s) repetida(s)	Abierto	Cerrado	Nombre
Sí	Sí	Sí		Camino
Sí	Sí		Sí	Camino (cerrado)
Sí	No	Sí		Recorrido
Sí	No		Sí	Circuito
No	No	Sí		Camino simple
No	No		Sí	Ciclo

Ejemplo de grafo 1



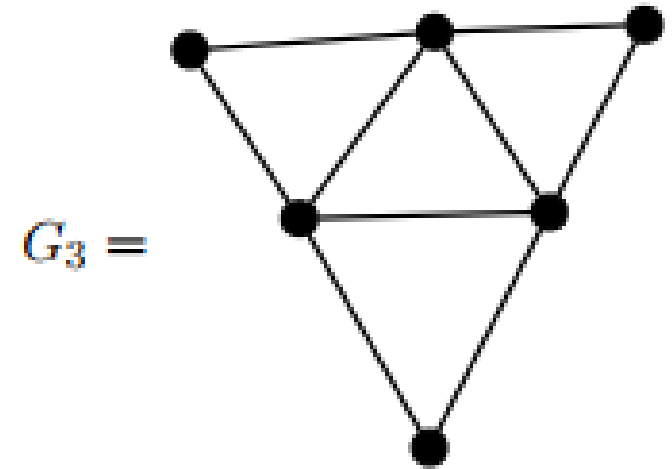
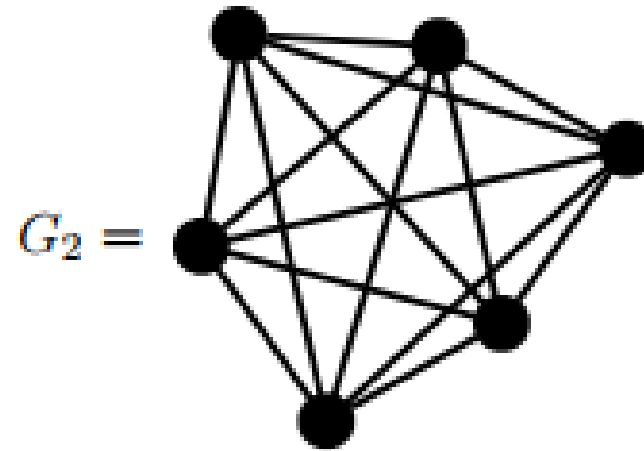
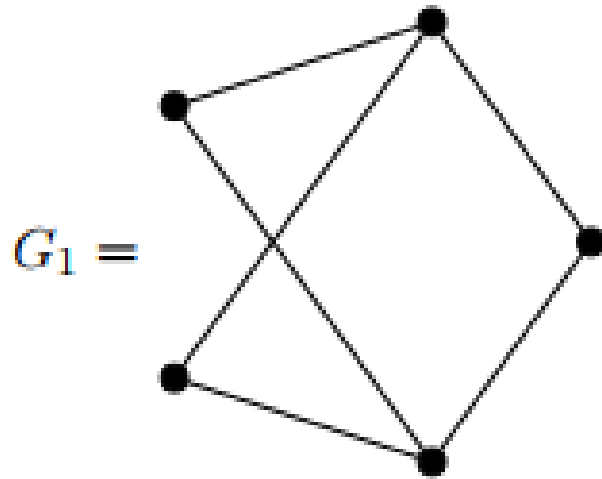
Ejemplo de grafo 3





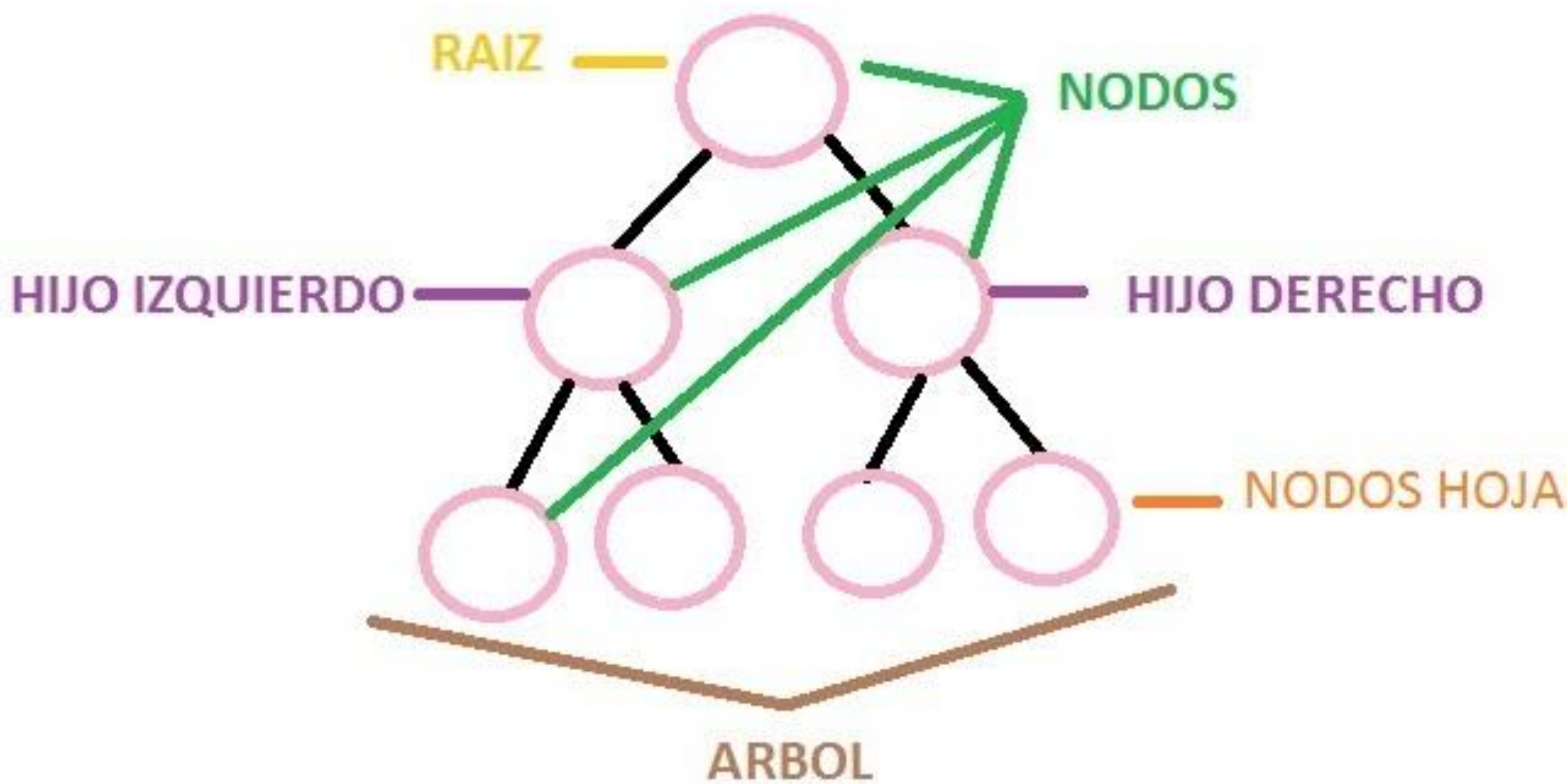
Subgrafos

Ejemplo de subgrafo



Arboles





Características de un árbol

- Es conexo.
- No contiene ciclos.
- El número de nodos es igual al número de vértices más uno.
- Si se le elimina una arista, queda desconectado en dos subgrafos de ese árbol que también serán árboles.





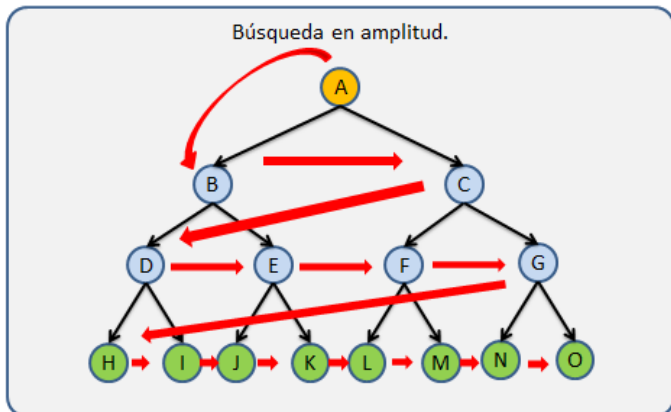
WWW.ULTRACOLORINGPAGES.COM

Algoritmos de búsqueda

Algoritmos de búsqueda

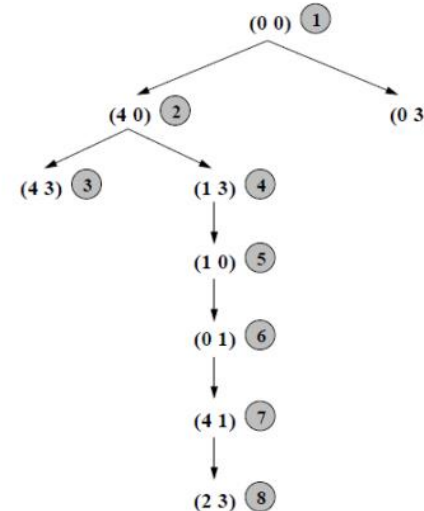
Búsqueda a lo ancho

Este algoritmo de búsqueda busca los caminos más cortos desde un vértice de origen dado a todos los demás vértices, en términos del número de aristas en los caminos



Búsqueda a lo largo

Este algoritmo de búsqueda intenta recorrer todos los nodos de un grafo o árbol de manera ordenada pero no uniforme, consistiendo en expandirse por todos los nodos que van encontrando en un camino en concreto



Algoritmo de Búsqueda a lo Ancho

Algoritmo de búsqueda en anchura

Paso 1: Insertamos el vértice v_1 en la cola Q e inicializamos T como el árbol formado por este único vértice v_1 (la raíz de la versión final de T).

Paso 2: Eliminamos los vértices del frente de Q . Al eliminar un vértice v , consideramos v_i para cada $2 \leq i \leq n$. Si la arista $\{v, v_i\} \in E$ y v_i no ha sido visitado, agregamos la arista a T . Si examinamos *todos* los vértices que ya estaban en Q y no obtenemos aristas nuevas, el árbol T (generado hasta ese momento) es el árbol recubridor (ordenado con raíz) del orden dado.

Paso 3: Insertamos los vértices adyacentes a cada v (del paso 2) en el final de la cola Q , según el orden en que fueron visitados por primera vez. Después regresamos al paso 2.

Algoritmo de búsqueda en profundidad

Paso 1: Se asigna v_1 a la variable v y se inicializa T como el árbol que consta solamente de este vértice. (El vértice v_1 será la raíz del árbol recubridor que se va a desarrollar.)

Paso 2: Seleccionamos el subíndice más pequeño i , $2 \leq i \leq n$, tal que $\{v, v_i\} \in E$ y v_i no ha sido visitado todavía.

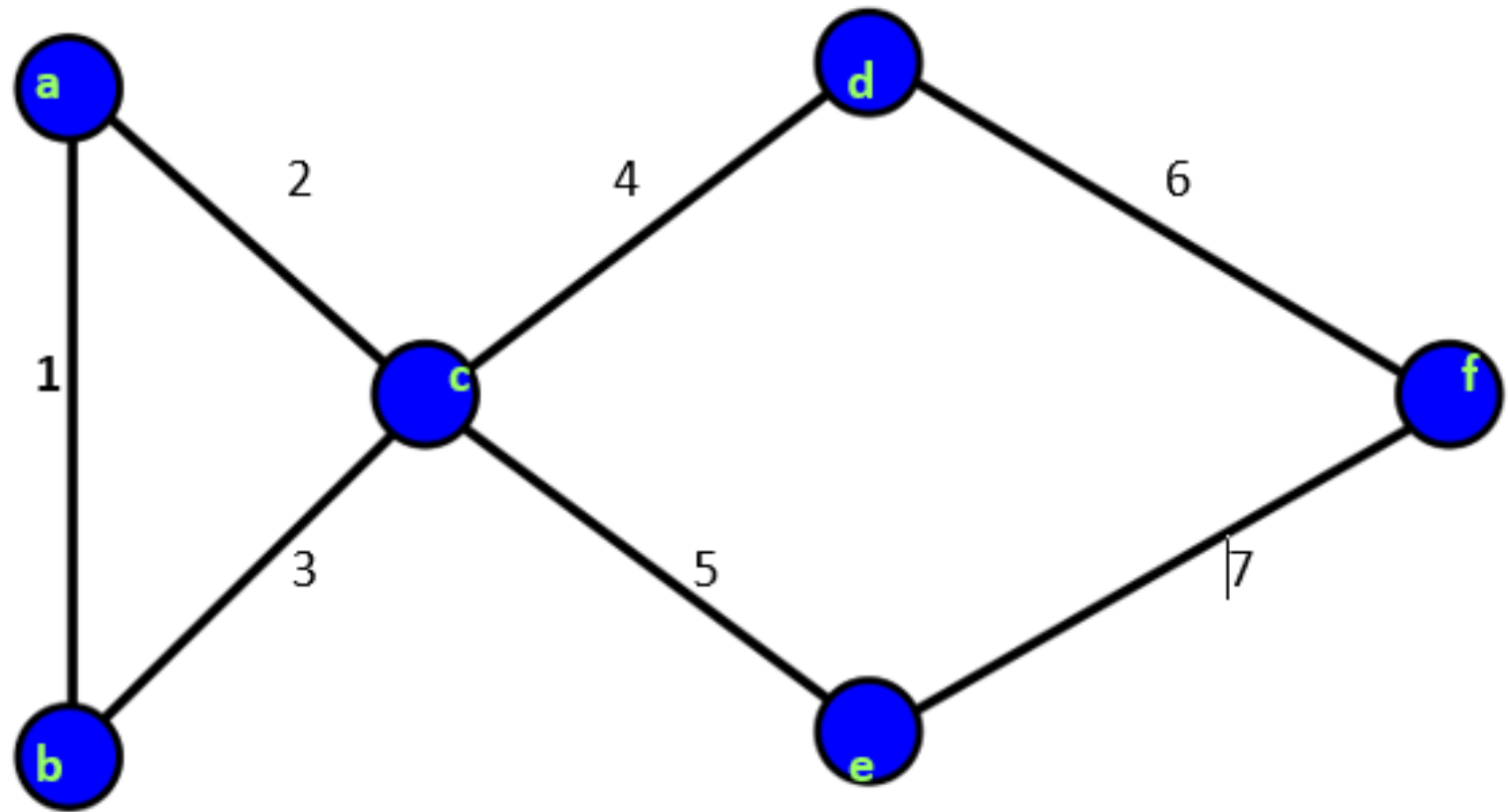
Si no se encuentra tal subíndice, entonces se va al paso 3. En caso contrario, se hace lo siguiente: (1) Añadimos la arista $\{v, v_i\}$ al árbol T ; (2) asignamos v_i a v ; y (3) regresamos al paso 2.

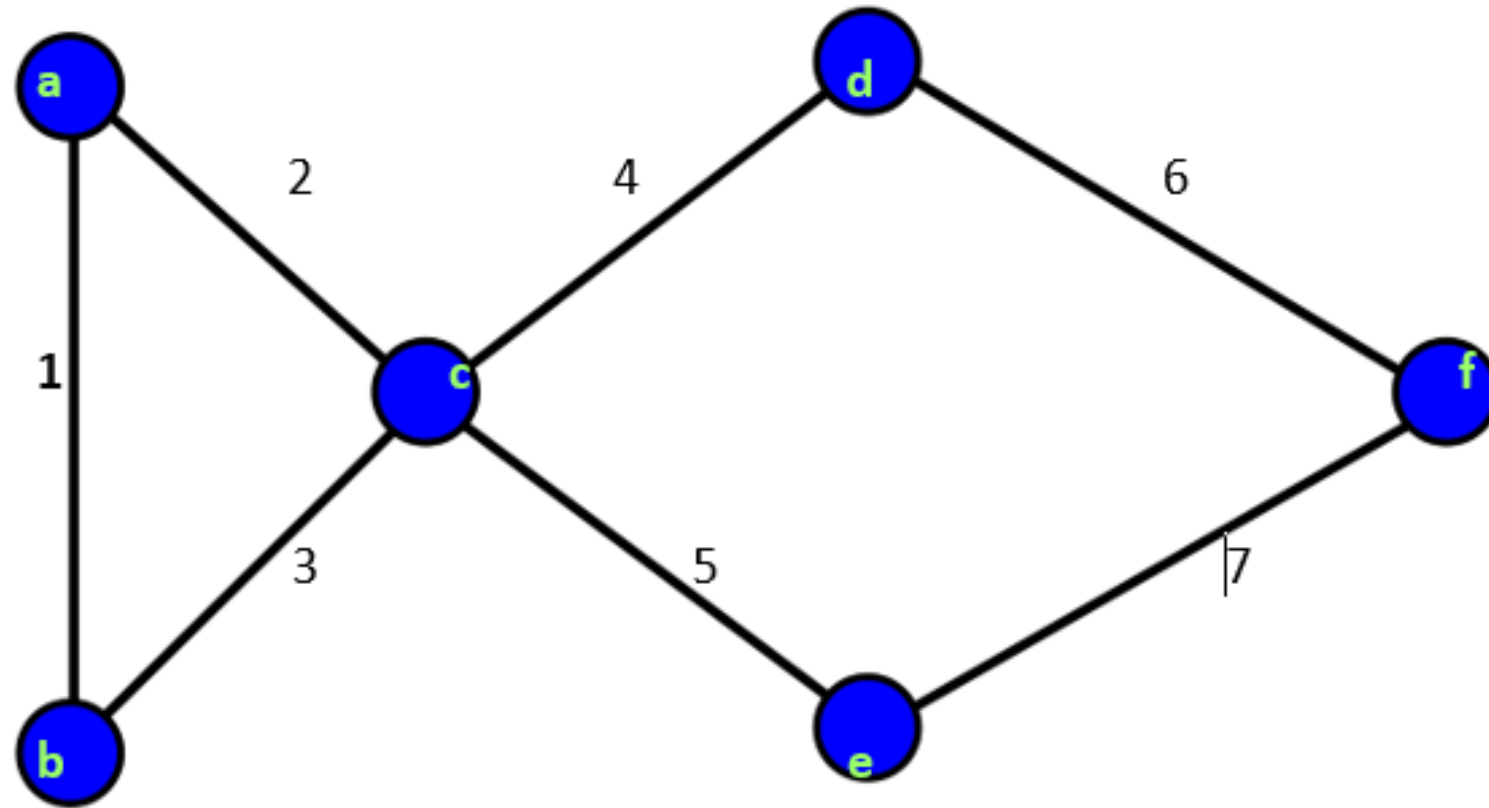
Paso 3: Si $v = v_1$, el árbol T es el árbol recubridor (ordenado, con raíz) del orden dado.

Paso 4: Si $v \neq v_1$, retrocedemos desde v . Si u es el padre del vértice asignado a v en T , entonces asignamos u a v y regresamos al paso 2.

Algoritmo de Búsqueda a lo Largo

Ejemplo de la búsqueda a la largo





Ejemplo de la búsqueda a lo ancho



Gracias

MIEMBROS DEL GRUPO 1:

Enner Mendizabal – 202302220

Daniel Castellanos – 202200176

Julio Ramos - 202200044

Eliot Ardón - 201901472