

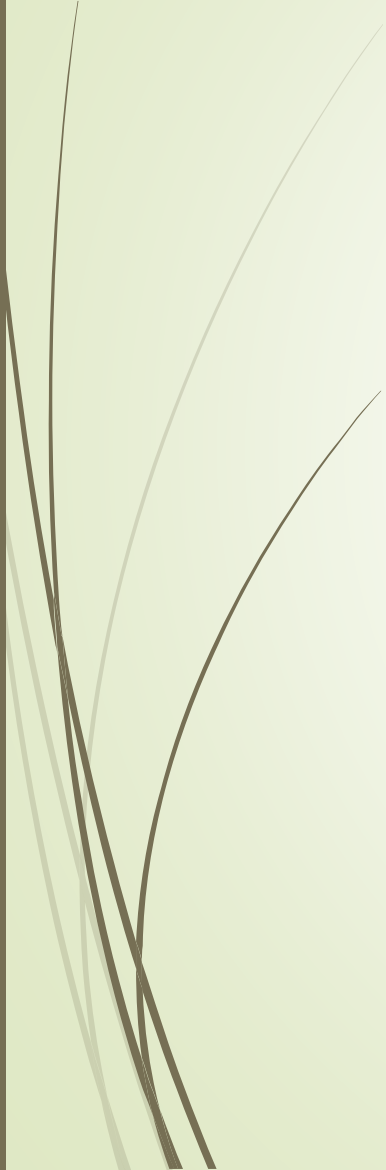
unidad 4

Estructuras no lineales

Árbol Binario

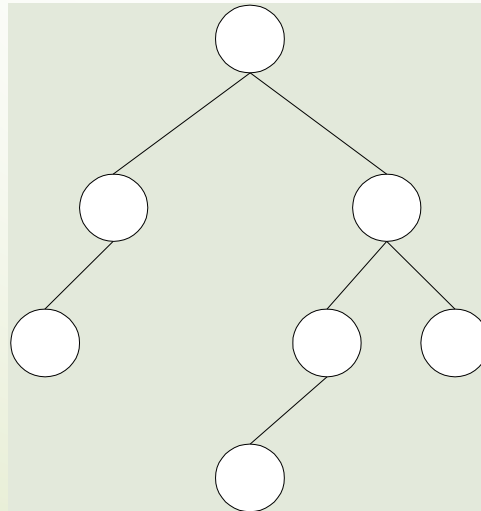
2

Arbol binario

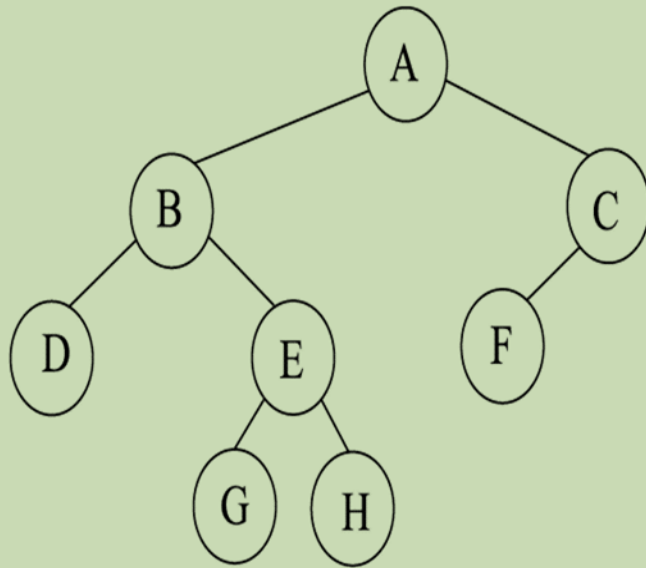


Arbol binario

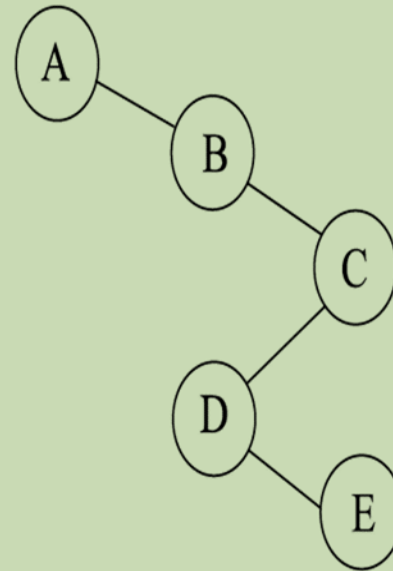
- Es un árbol en el que todos sus nodos a lo mas tienen dos hijos (hijo izquierdo o subárbol izquierdo, e hijo derecho o subárbol derecho).
- Los arboles binarios pueden usarse para representar una estructura en la cual es posible tomar decisiones con dos opciones.



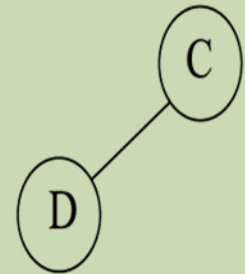
Arboles binarios



a)



b)

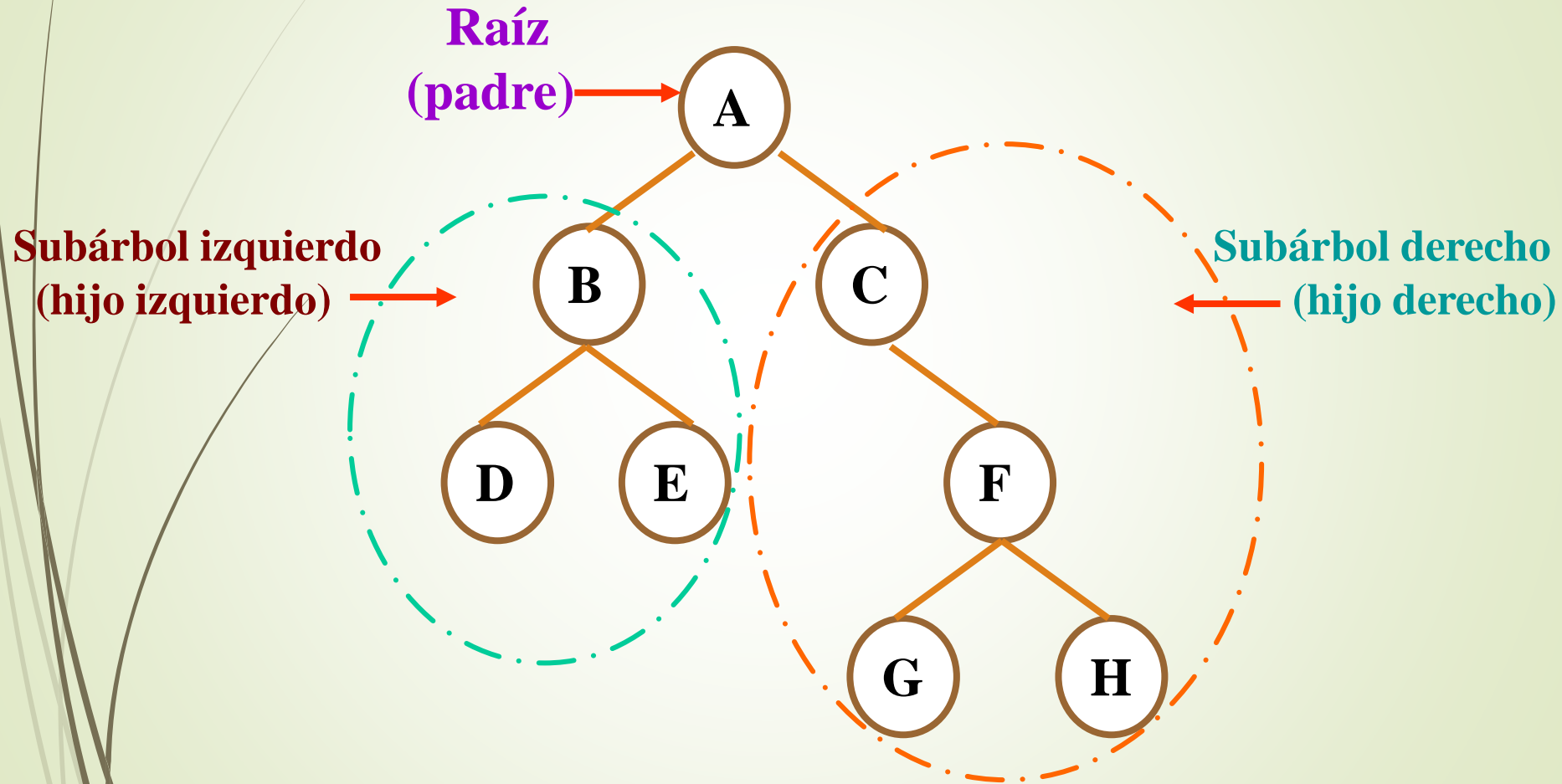


c)

Árbol binario – definición recursiva

Es un conjunto finito de elementos que puede estar vacío o contener un elemento denominado la **raíz** o **padre** del árbol y otros elementos divididos en dos subconjuntos separados, cada uno de los cuales es en sí un **árbol binario**. Estos dos subconjuntos son denominados **subárbol izquierdo** o **hijo izquierdo** y **subárbol derecho** o **hijo derecho**.

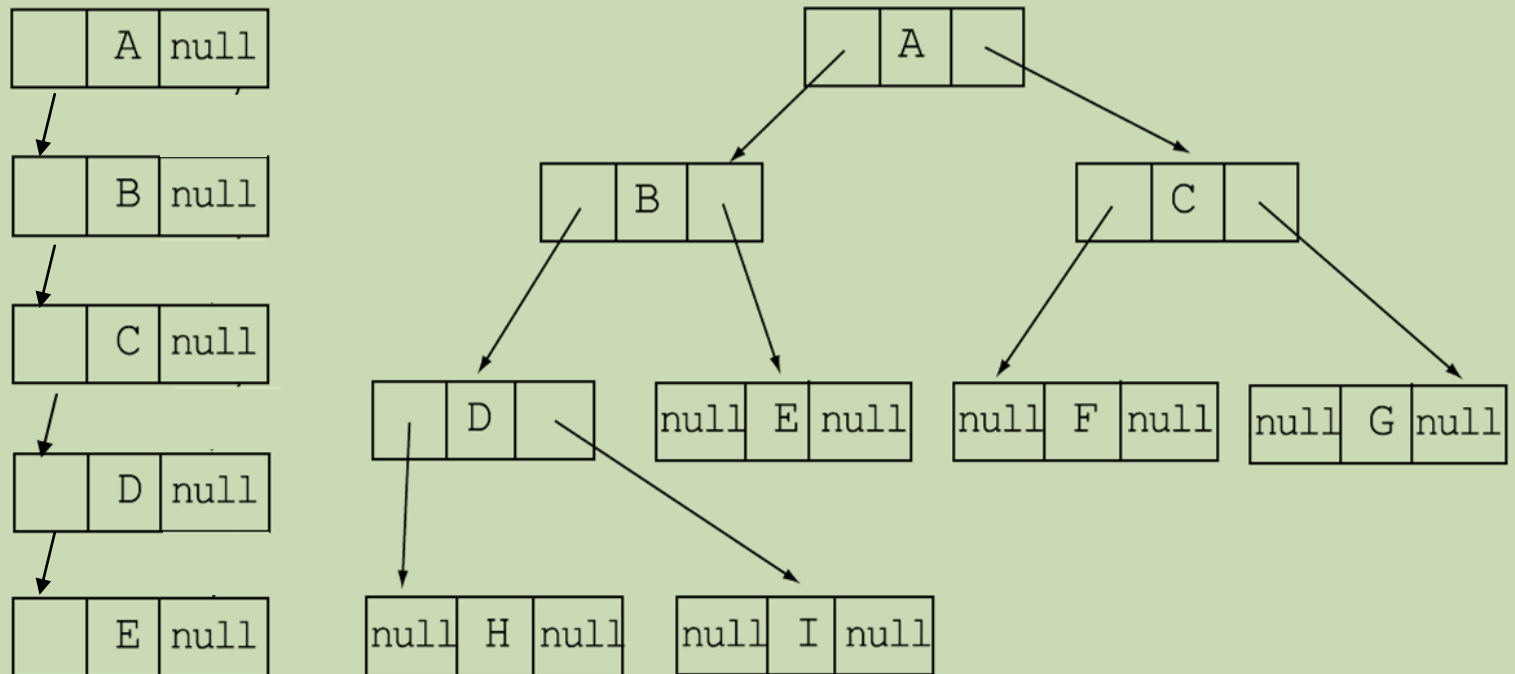
Representación



ESTRUCTURA DE UN ÁRBOL BINARIO

7

- Un árbol binario se construye con nodos. Cada nodo debe contener el campo *dato* (datos a almacenar)
- y dos campos de enlace (*apuntador*), uno al subárbol izquierdo (**izquierdo, izdo**) y otro al
- subárbol derecho (**derecho, dcho**). El valor null indica un árbol o un subárbol vacío



Representación enlazada de dos árboles binarios

Operaciones básicas que definen el TAD árbol binario

8

- *CrearArbol* Inicia el árbol como vacío.
- *Construir* Crea un árbol con un elemento raíz y dos ramas, izquierda y derecha que son a su vez árboles.
- *EsVacio* Comprueba si el árbol no tiene nodos.
- *Raiz* Devuelve el nodo raíz.
- *Izquierdo* Obtiene la rama o subárbol izquierdo de un árbol dado.
- *Derecho* Obtiene la rama o subárbol derecho de un árbol dado.
- *Borrar* Elimina del árbol el nodo con un elemento determinado.
- *Pertenece* Determina si un elemento se encuentra en el árbol.

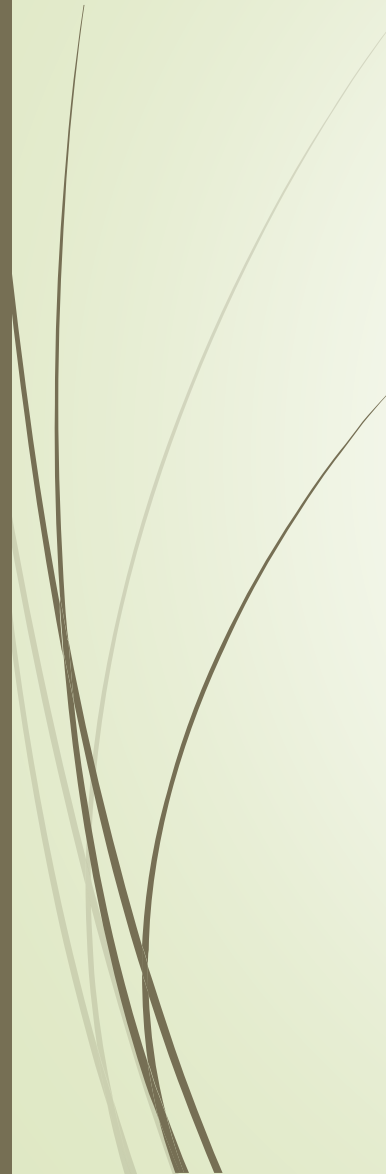
Operaciones en árboles binarios

- Algunas de las operaciones típicas que se realizan en árboles binarios son las siguientes:
- Determinar su altura.
- Determinar su número de elementos.
- Hacer una copia.
- Visualizar el árbol binario en pantalla o en impresora.
- Determinar si dos árboles binarios son idénticos.
- Borrar (eliminar el árbol).
- Si es un árbol de expresión, evaluar la expresión.
- Todas estas operaciones se pueden realizar recorriendo el árbol binario de un modo sistemático.

Recorridos de un árbol Binario

El recorrido es la operación de visita al árbol o, lo que es lo mismo, la visita a cada nodo del árbol una vez y sólo una.

- **Recorrido en preorden (prefijo).**
- **Recorrido en inorden (infijo).**
- **Recorrido en postorden (postfijo).**



Recorridos de un árbol Binario

RID

- **Recorrido en preorden (prefijo)**
 - Visita la raíz.
 - Recorre el subárbol izquierdo en preorden.
 - Recorre el subárbol derecho en preorden.

Recorridos de un árbol Binario

preorden (NODO)

{NODO es un apuntador a registro}

Si **NODO** \neq null entonces

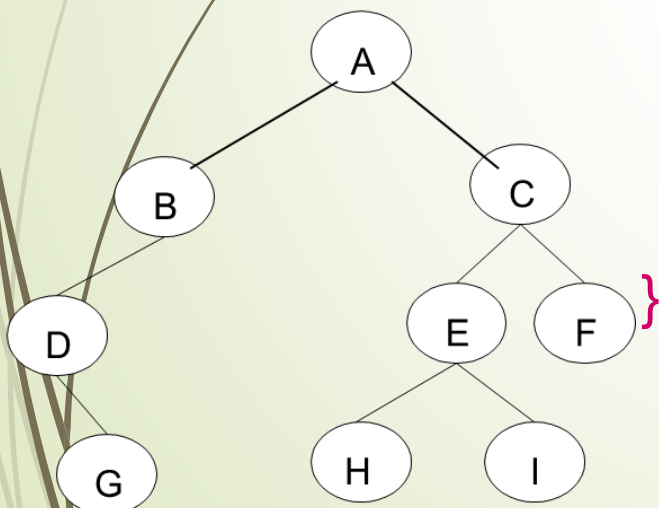
{

Escribe (NODO.INFO);

preorden (NODO.IZQ);

preorden (NODO.DER);

}



Preorden = A B D G C E H I F

Recorridos de un árbol Binario

IRD

- **Recorrido en inorden (infijo)**
 - Recorre el subárbol izquierdo en inorden.
 - Visita la raíz
 - Recorre el subárbol derecho en inorden.

Recorridos de un árbol Binario

inorden (NODO)

{NODO es un apuntador a registro}

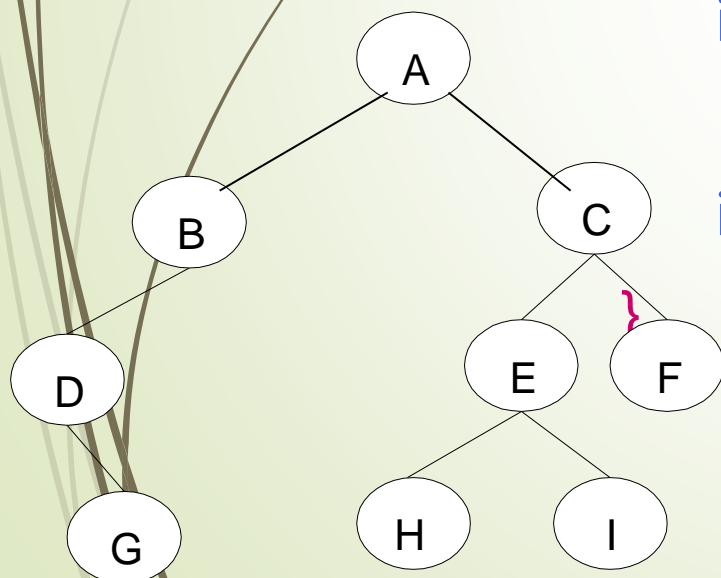
Si NODO != null entonces

{

inorden (NODO.IZQ);

Escribe NODO.INFO;

inorden (NODO.DER);



Inorden: D G B A H E I C F

Recorridos de un árbol Binario

IDR

- **Recorrido en postorden (postfijo)**
 - Recorre el subárbol izquierdo en postorden.
 - Recorre el subárbol derecho en postorden.
 - Visita la raíz.

Recorridos de un árbol Binario

postorden (NODO)

Si NODO != null entonces

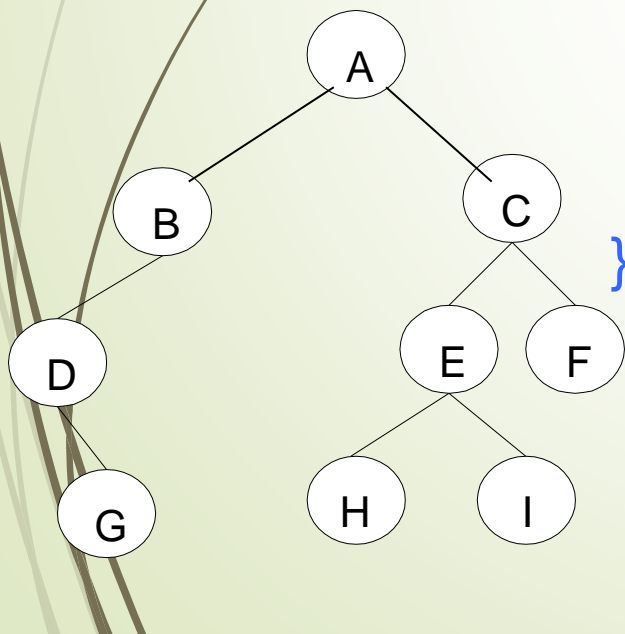
{

postorden (NODO.IZQ);

postorden (NODO.DER);

Escribe NODO.INFO;

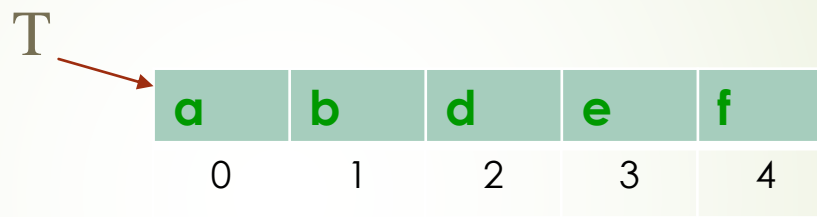
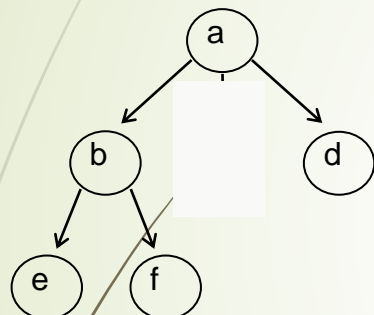
}



Postorden : G D B H I E F C A

Representación en memoria de árboles binarios

- 1. memoria estática: por medio de arreglos



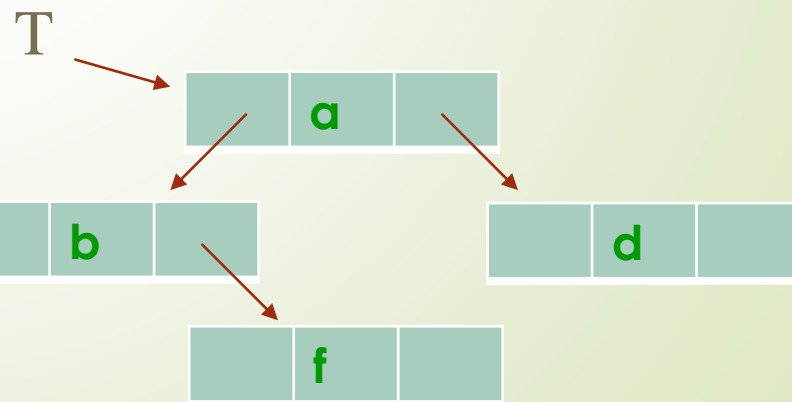
- 2. memoria dinámica: objetos llamados nodos



información
del nodo

dirección del
subárbol derecho

dirección del
subárbol izquierdo





Referencias



- Estructuras de datos, Osvaldo Cairo, Silvia guardati. Ed Mc Graw-Hill
- Estructura de datos en C++, Dr. Romeo Sánchez Nigenda.
- <https://www.centroestudioscervantinos.es/fundamentos-y-aplicaciones-de-la-teoria-de-los-grafos/>