

Arboles

Algoritmos y Estructuras de Datos

UNLA

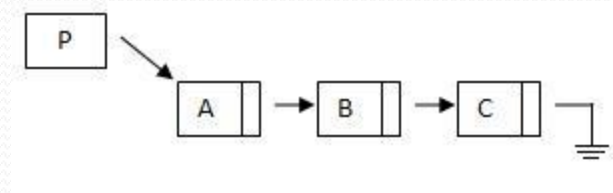
Lic. Alejandro Sasin

Diego Cañete

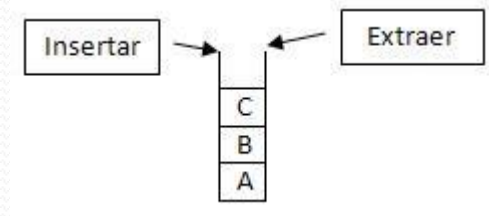
Introducción

- Hasta el momento vimos las siguientes estructuras de datos:

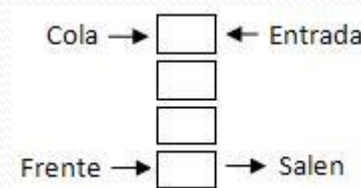
- Listas



- Pilas

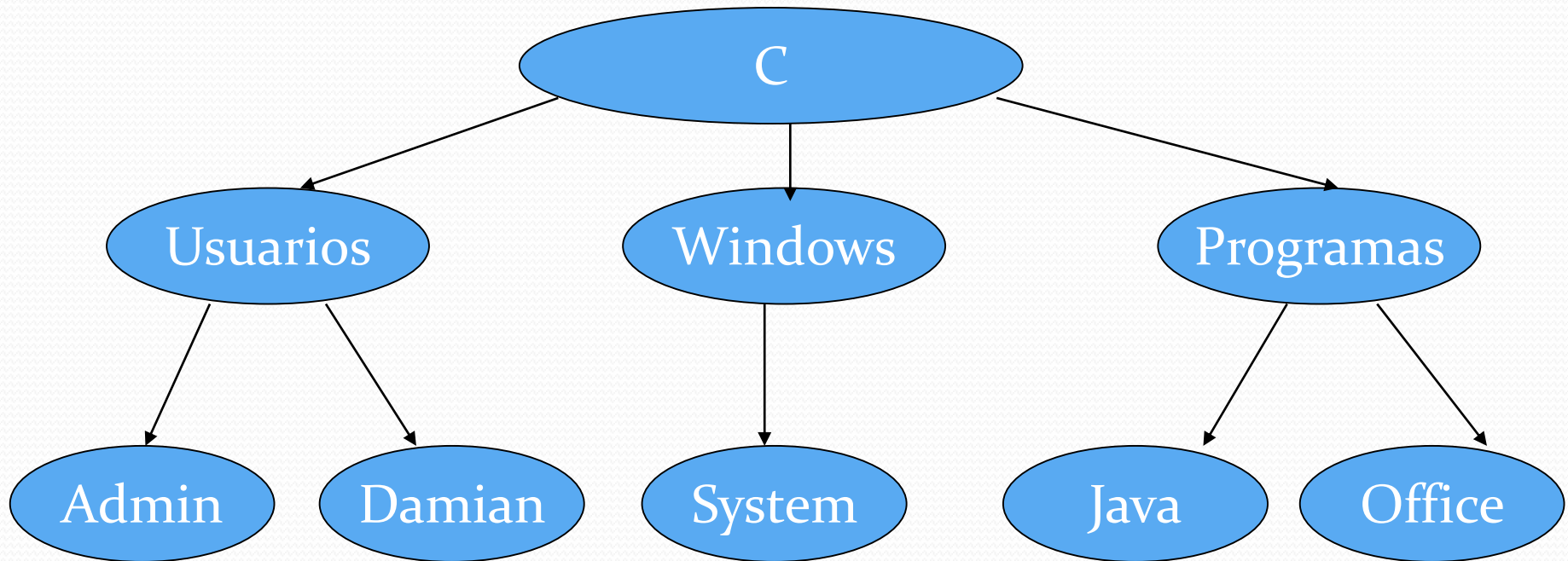


- Colas



Problema

- Cómo representaríamos la siguiente estructura?



Arboles

- Un **árbol** (tree) es un T.D.A. que consta de un conjunto finito T de nodos y una relación R (paternidad) entre los nodos tal que:
 - Hay un nodo, especialmente designado, llamado la **raíz** del árbol T .
 - Los nodos restantes, excluyendo la raíz, son particionados en m ($m \geq 0$) conjuntos disjuntos T_1, T_2, \dots, T_m , cada uno de los cuales es, a su vez, un árbol, llamado **subárbol** de la raíz del árbol T .
 - A los nodos que no son raíces de otros subárboles se les denomina **hojas** del árbol T , o sea, no tienen **sucesores** o **hijos**.

Definiciones

Padre de un nodo al nodo raíz del subárbol más pequeño que contiene a dicho nodo y en el cual él no es raíz.

Hijo de un nodo al (los) nodo(s) raíz(ces) de uno de sus subárboles.

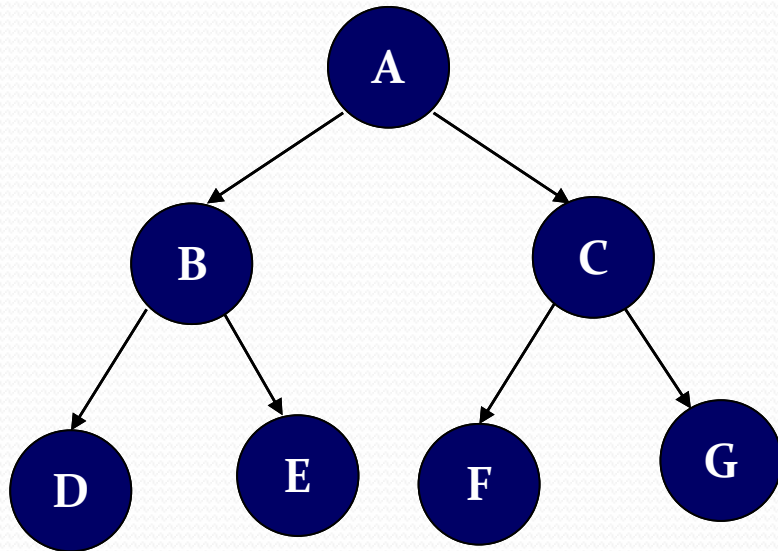
Predecesor de un nodo al nodo que le antecede en un recorrido del árbol.

Hermano de un nodo a otro nodo hijo de su padre.

Aclaraciones

- Si el conjunto finito T de nodos del árbol es vacío, entonces se trata de un ***árbol vacío***.
- En esta estructura existe **sólo un nodo** sin padre, que es la ***raíz*** del árbol.
- Todo nodo, a excepción del nodo raíz, tiene ***uno y sólo un padre***.
- Los subárboles de un nodo son llamados ***hijos***.

Ejemplos



Padre de C:

A

Padre de E:

B

Padre de G:

C

Padre de A:

NO

Hijos de A:

B

C

Hijos de C:

F

G

Hijos de F:

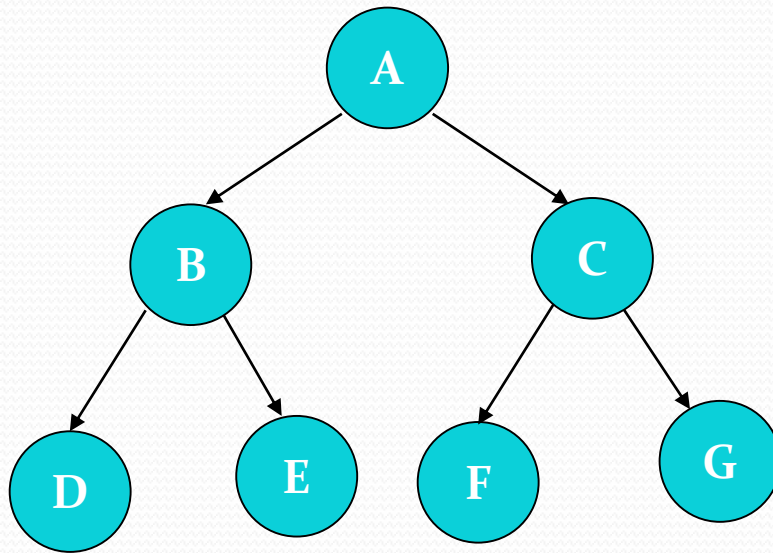
NO

Aclaración

- Para todo nodo k , distinto de la raíz, existe una única secuencia de la forma:
 - $k_0, k_1, k_2, k_3, \dots, k_n$, donde k_0 =raíz y $k_n=k$

Con $n \geq 1$, donde k_i es el sucesor de k_{i-1} , para $1 \leq i \leq n$, o sea, cada nodo k_i de la secuencia es la raíz de otro subárbol.

Ejemplo



Secuencias

de A a G: A-C-G

de A a E: A-B-E

de B a F: No existe

Definiciones

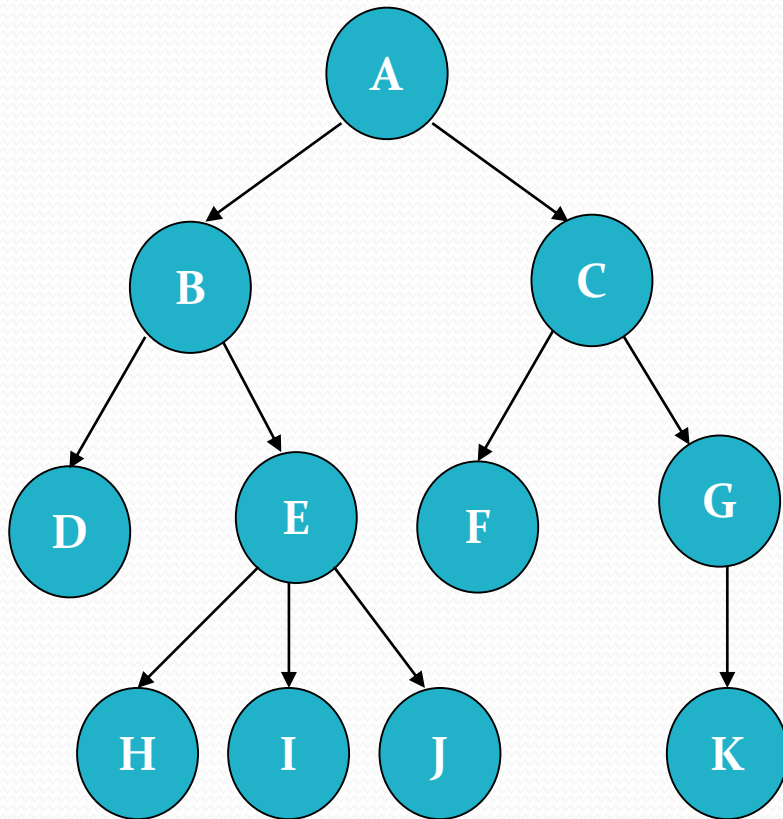
Grado de un nodo: cantidad de hijos de un nodo.

Grado de un árbol: el mayor de los grados de todos sus nodos.

Nodo hoja: es un nodo sin hijos, es decir, con grado = 0.

Nodo rama: es un nodo que tiene hijos, o sea, a la raíz de un subárbol (grado > 0).

Ejemplo



Grado

de A:

2

de E:

3

de G:

1

de J:

0

Grado del árbol:

3

Nodos hojas:

D, H, I, J, F, K

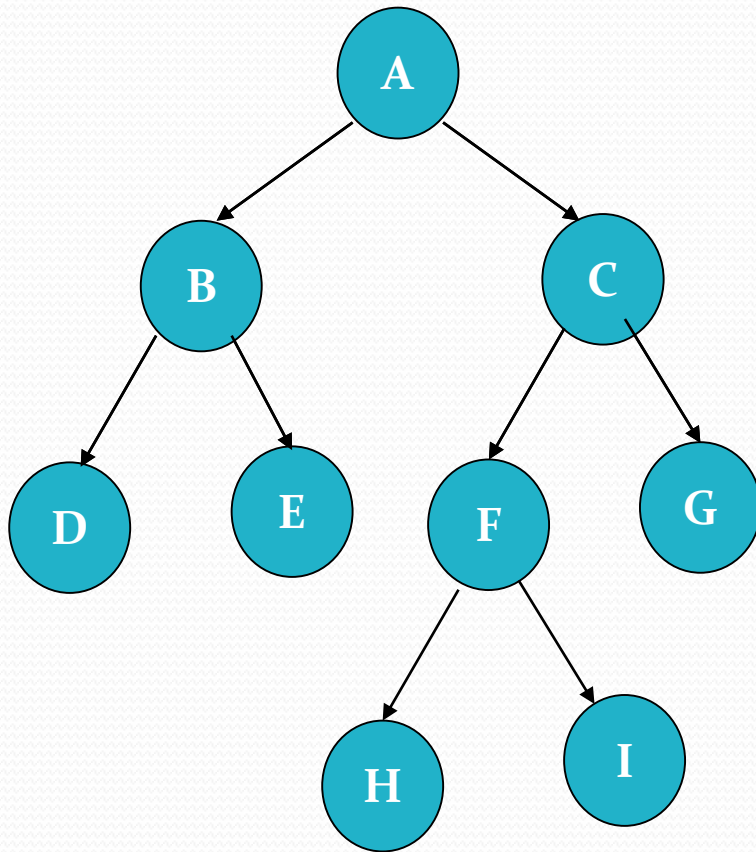
Nodos ramas:

A, B, C, E, G

Definiciones

- *Nivel de un nodo* al nivel de su padre más uno. Por definición, la raíz del árbol tiene nivel 0. Esta definición es recursiva.
- La **profundidad** de un árbol es el máximo nivel de cualquier hoja en el árbol. Esto es igual a la longitud de la trayectoria más larga de la raíz a cualquier hoja.

Ejemplo



Nivel

de A:

0

de E:

2

de B:

1

de I:

3

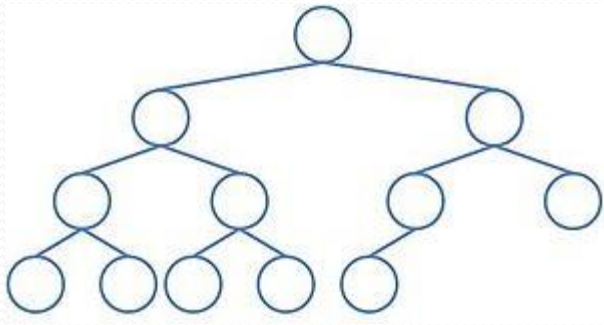
de G:

2

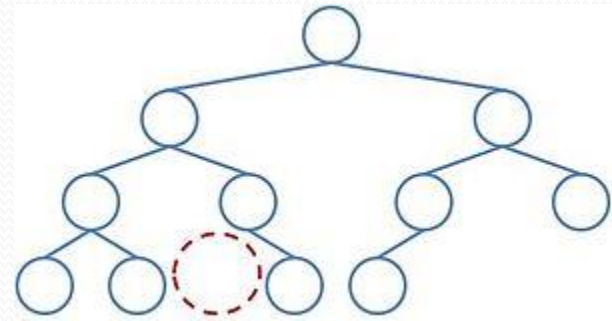
Definición

- **Árbol completo de nivel n** Un árbol completo es un árbol de profundidad K que tiene todos los nodos posibles hasta el penúltimo nivel (profundidad $K-1$), y donde los elementos del último nivel están colocados de izquierda a derecha sin dejar huecos entre ellos.

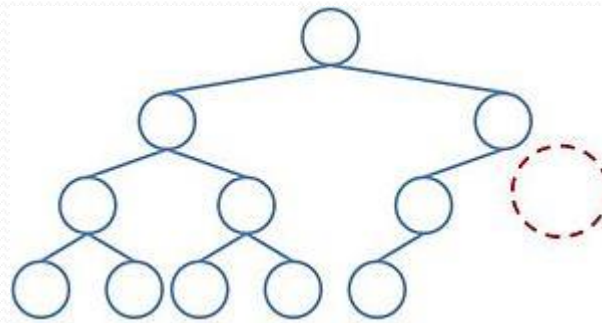
Ejemplo



completo



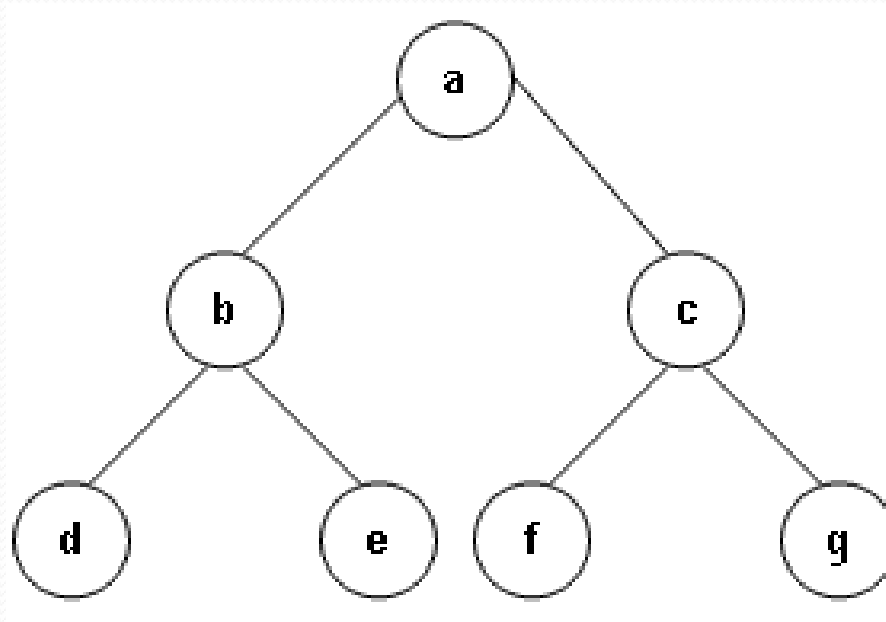
No completo



No completo

Definición

- *Árbol lleno: es un árbol con todos sus niveles llenos.*



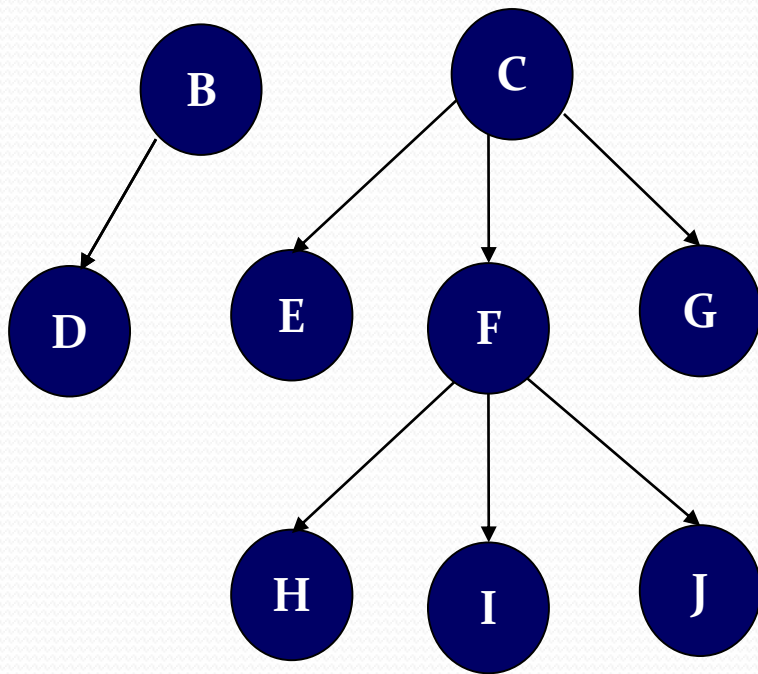
Definición

Una *foresta* es una colección de dos o más árboles disjuntos.

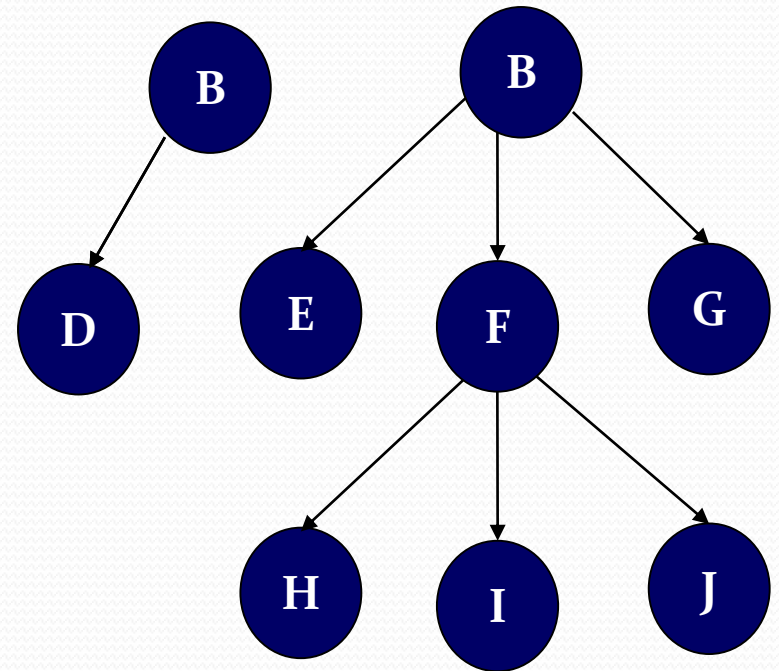
Aclaraciones:

- Disjuntos significa que no hay nodos en común entre dos árboles cualesquiera de la misma.
- De un árbol se obtiene una foresta al quitarle la raíz, si tiene dos hijos o más.
- De una foresta se obtiene un árbol al añadir un nodo que sea raíz de todos los árboles que la conforman.

Ejemplo



Es una floresta



NO es una floresta

Definición

- Árbol ordenado: todo árbol para el que se considera el orden relativo de los sucesores o subárboles de cualquier nodo.
- Es decir, en un árbol ordenado se habla de primero, segundo o último hijo de un nodo en particular. El primer hijo de un nodo de un árbol ordenado es denominado el hijo mayor de ese nodo y el último hijo es denominado el menor.
- El Árbol es ordenado si al intercambiar el orden relativo de los subárboles de un nodo, representa una situación semánticamente diferente.

Definición

Árbol orientado a un árbol para el cual no interesa el orden relativo de los sucesores o subárboles de cualquier nodo, ya que sólo se tiene en cuenta la orientación de los nodos.

Ejemplo:

La estructura organizativa de una empresa, donde no es importante el orden de los subdirectores a la hora de representarlos en el árbol.

En la solución de problemas informáticos, los más utilizados son los árboles ordenados.

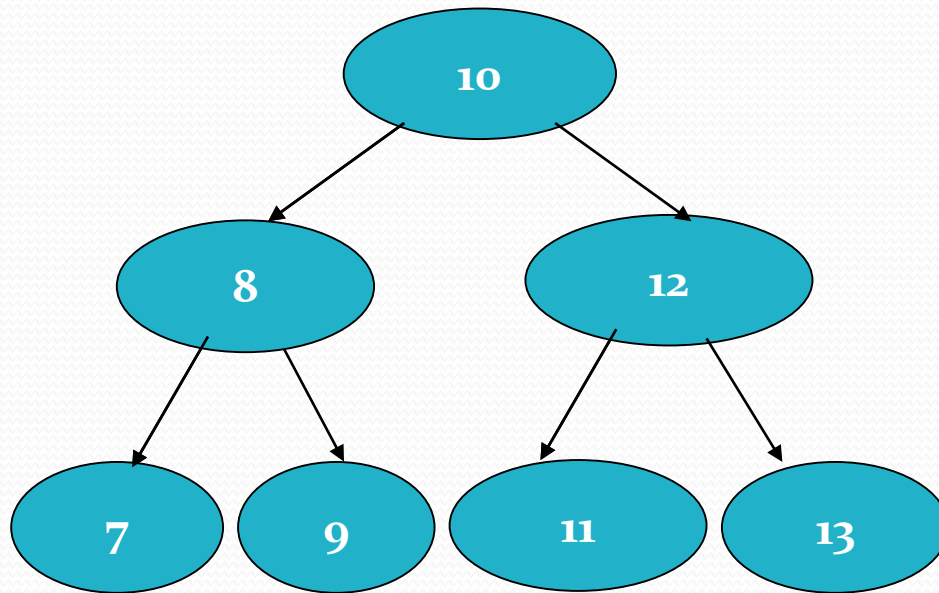
Arbol Binario

Un *árbol binario* (en inglés *binary tree*) es un árbol ordenado de grado 2.

Aclaraciones:

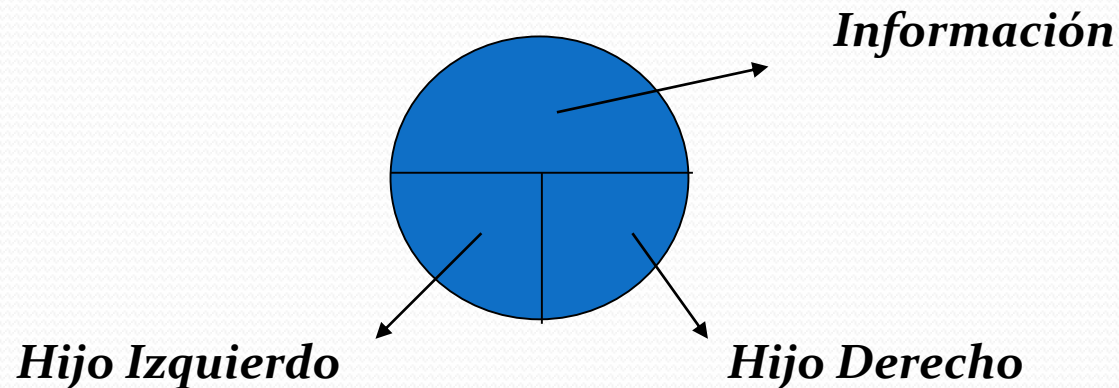
- Grado 2 significa que cada nodo tiene como máximo dos hijos, o sea, dos subárboles.
- Al ser ordenado el árbol, importa el orden de los subárboles, es decir, que será necesario especificar de cada nodo cuál es el hijo izquierdo y cuál el hijo derecho.

Ejemplo



Implementación

- Cada nodo del árbol binario contiene:
 - Una referencia a su información.
 - Un puntero a su hijo izquierdo.
 - Un puntero a su hijo derecho.



Recorridos de un árbol

Los recorridos se clasifican de acuerdo al momento en que se visita la raíz del árbol y los subárboles izquierdo y derecho.

Existen tres recorridos:

- Recorrido en Preorden
- Recorrido en orden simétrico o inorden
- Recorrido en orden final o Postorden

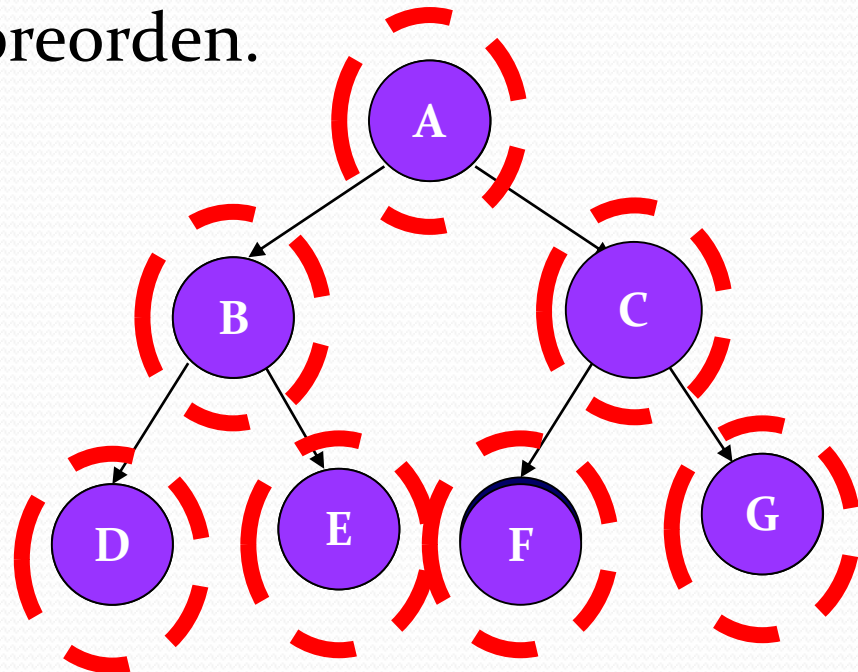
Recorrido en PreOrden

1. Visitar la raíz.
 2. Recorrer subárbol izquierdo en preorden.
 3. Recorrer subárbol derecho en preorden.
- Ejemplo Uso: Se va a utilizar siempre que queramos comprobar alguna propiedad del árbol (p.ej.: localizar elementos)

Recorrido en PreOrden

Recorrido:

1. Raíz.
2. Subárbol izquierdo en preorden.
3. Subárbol derecho en preorden.



* Las letras en los nodos identifican al nodo y no al valor que contiene.

Recorrido InOrden (simétrico)

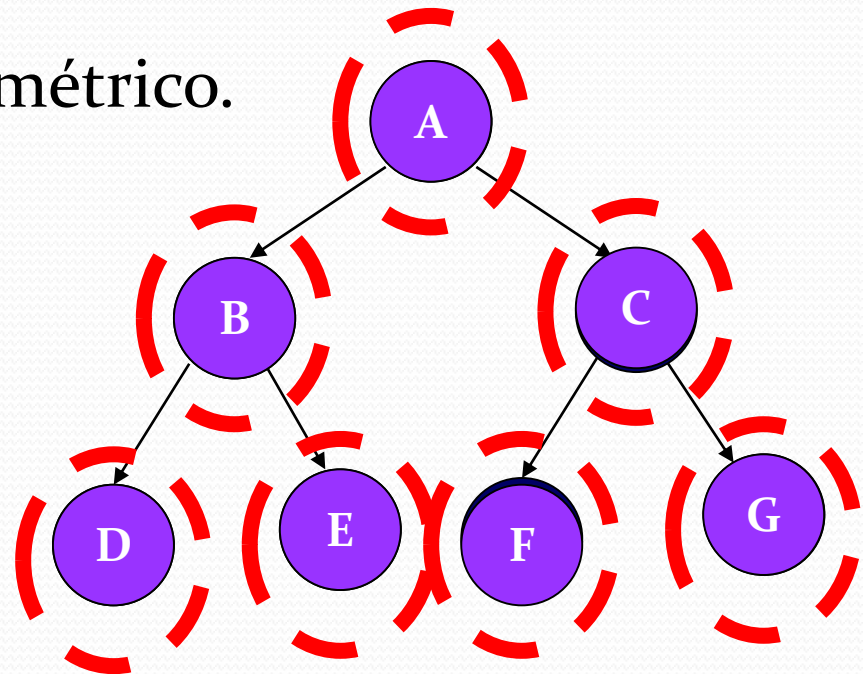
1. Recorrer subárbol izquierdo en simétrico.
 2. Visitar la raíz.
 3. Recorrer subárbol derecho en simétrico.
- Ejemplo Uso: Se utiliza siempre que nos pidan algo referido a la posición relativa de las claves o algo que tenga que ver con el orden de las claves (p.ej.: ¿Cuál es la 3ª clave?)

Recorrido InOrden

Recorrido

1. Subárbol izquierdo en simétrico.
2. Raíz.
3. Subárbol derecho en simétrico.

D	B	E	A	F	C	G
---	---	---	---	---	---	---



* Las letras en los nodos identifican al nodo y no al valor que contiene.

Recorrido PostOrden

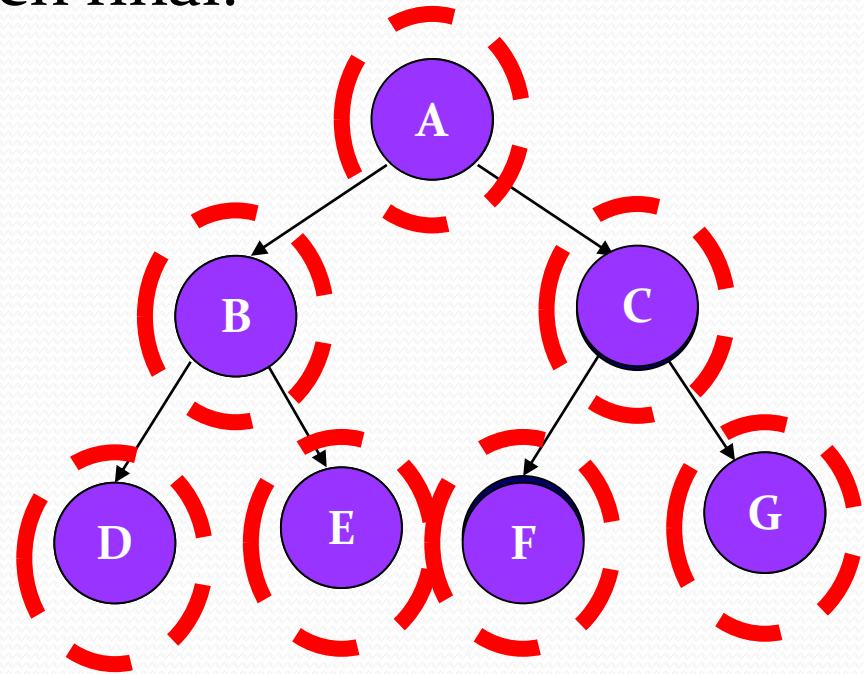
1. Recorrer subárbol izquierdo en orden final.
2. Recorrer subárbol derecho en orden final.
3. Visitar la raíz.
4. Ejemplo Uso: Su principal utilidad consiste en liberar la memoria ocupada por un árbol.

Recorrido PostOrden

Recorrido

1. Subárbol izquierdo en orden final.
2. Subárbol derecho en orden final.
3. Raíz.

D	E	B	F	G	C	A
---	---	---	---	---	---	---



* Las letras en los nodos identifican al nodo y no al valor que contiene.

Arbol de búsqueda

Permiten realizar operaciones (recorridos, búsqueda de un elemento, etc) de forma más eficiente.

Hay dos momentos para la manipulación de un árbol:

- La construcción del árbol.
- El recorrido del árbol para realizar las operaciones requeridas según el problema a resolver.

Existen dos tipos especiales de árboles:

- Árboles lexicográficos.
- Árboles hilvanados.

Arbol Lexicográfico

Un *árbol lexicográfico* es un árbol binario que, recorrido en orden simétrico, permite obtener la información de los nodos en algún criterio de ordenamiento.

La técnica de construcción de un árbol lexicográfico consiste en un proceso recursivo que va colocando los nodos en el subárbol izquierdo o derecho del nodo raíz, según sea el criterio de ordenamiento deseado (ascendente o descendente).

Arbol Lexicográfico

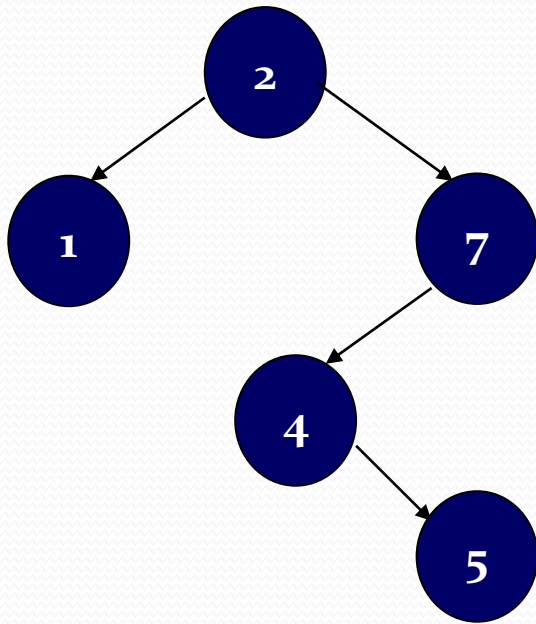
Siguiendo un ordenamiento ascendente:

1. Se compara el nodo que se quiere insertar con la raíz del árbol.
 - Si es menor, se coloca en el subárbol izquierdo siguiendo el mismo proceso.
 - Si es mayor, se coloca en el subárbol derecho siguiendo el mismo proceso.

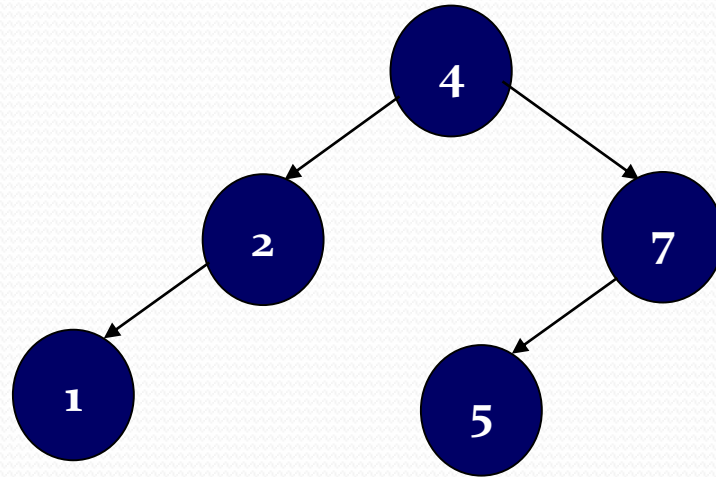
Ejemplo

- Árbol lexicográfico con ordenamiento ascendente.

Lista: 2, 7, 1, 4, 5



Lista: 4, 7, 2, 1, 5



Si se recorre en orden simétrico, se obtiene la información de sus nodos en orden ascendente: 1, 2, 4, 5, 7 con independencia del orden de la lista original.

Situación

- El recorrido de árboles con programas recursivos resulta costoso ya que implica un gasto adicional de memoria y tiempo de ejecución. Para árboles muy grandes se puede desbordar el stack del sistema relativamente pronto.
- → árbol hilvanado

Árbol Hilvanado

- Un *árbol hilvanado* (o *árbol entrelazado*) es un árbol binario en el que cada hijo izquierdo de valor nulo es sustituido por un enlace o hilván al nodo que le antecede en orden simétrico (excepto el primer nodo en orden simétrico) y cada hijo derecho de valor nulo es sustituido por un enlace o hilván al nodo que le sigue en el recorrido en orden simétrico (excepto el último nodo en orden simétrico).

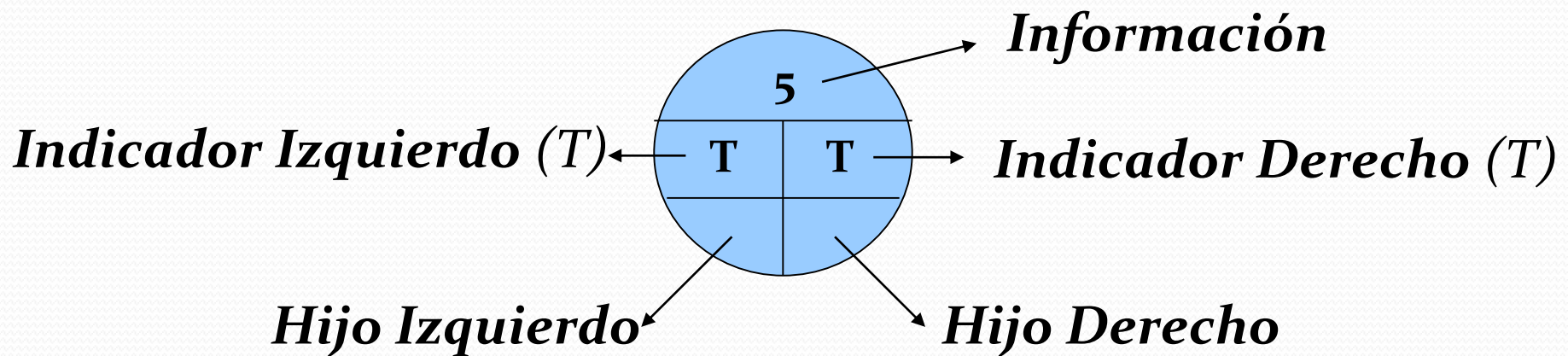
Árbol Hilvanado

Ahora, un recorrido en orden simétrico se puede implementar sin necesidad de recursión.

Sin embargo, se requiere que los nodos tengan en su estructura algún atributo que permita saber cuándo un enlace es real y cuándo se trata de un hilván. En este caso es necesario un atributo para cada hijo.

Árbol Hilvanado

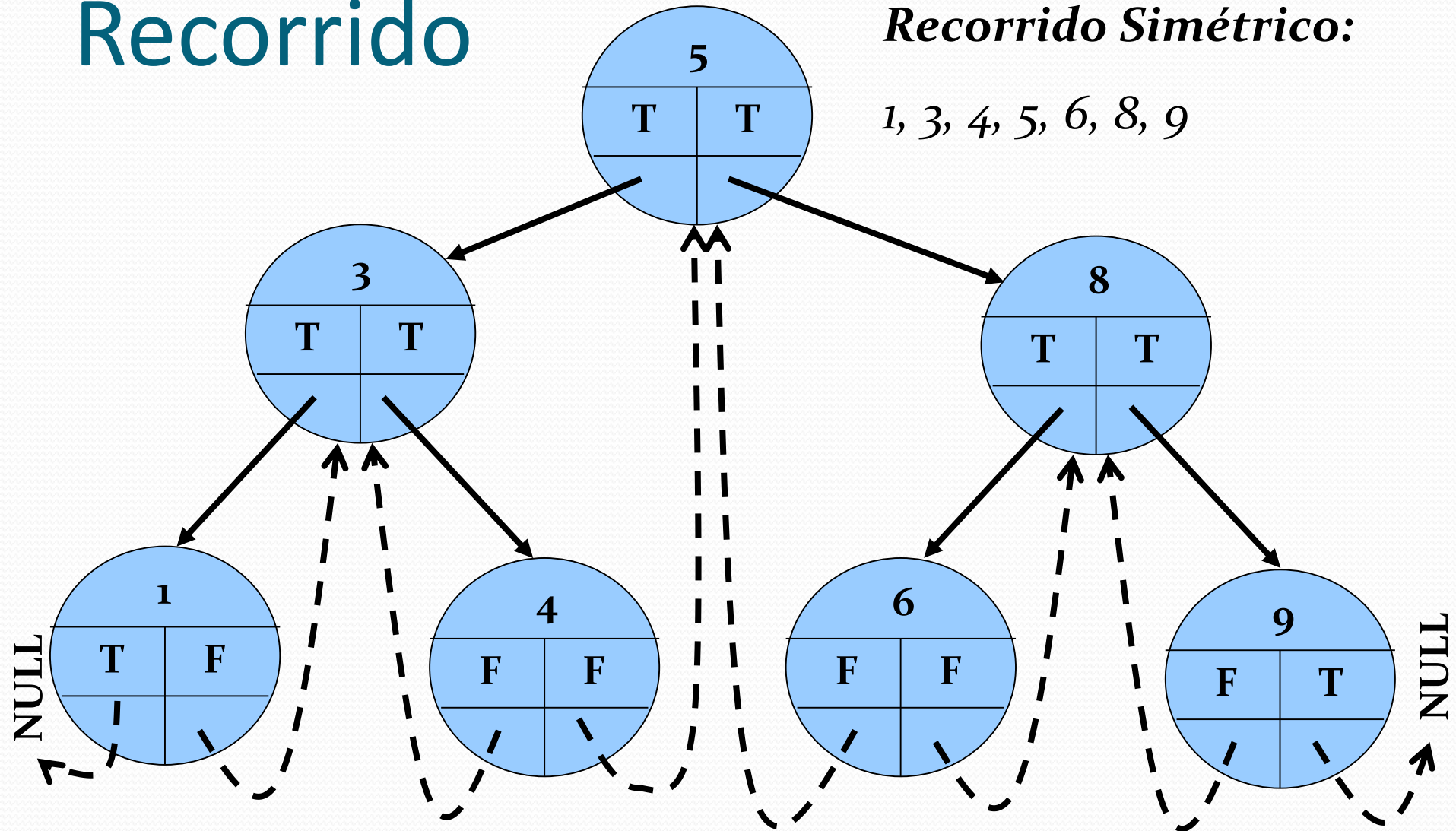
- Cada nodo del árbol hilvanado contiene:
 - Una referencia a su información.
 - Un apuntador a su hijo izquierdo.
 - Indicador Izquierdo (Verdadero o Falso).
 - Un apuntador a su hijo derecho.
 - Indicador Derecho (Verdadero o Falso).



Recorrido

Recorrido Simétrico:

1, 3, 4, 5, 6, 8, 9



Árbol Balanceado

La búsqueda en un árbol puede convertirse en una búsqueda secuencial. Esto sucede porque el árbol no está balanceado, es decir los nodos no están distribuidos uniformemente y se han insertado todos los nodos en profundidad.

Esto podría ser salvado si se utilizara un árbol balanceado que al insertar toma en cuenta la cantidad de niveles del árbol y distribuye los nodos uniformemente.

Arbol AVL

Un *árbol AVL* es un árbol binario de búsqueda en el que las alturas de los subárboles izquierdo y derecho de cualquier nodo se diferencian a lo sumo en uno.

La búsqueda es similar a como se hace en un árbol binario de búsqueda (lexicográficos), pero la inserción y la eliminación deben considerar la propiedad del balance.



Fin

Dudas, consultas, sugerencias?