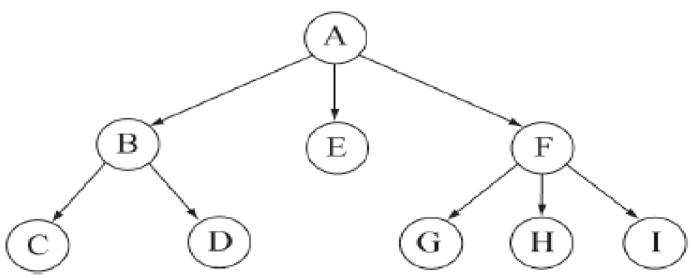
CAPÍTULO 2. ÁRBOLES

2.0. CONCEPTOS GENERALES

Los árboles (en general) se utilizan para representar fórmulas algebraicas, para organizar objetos en orden de tal forma que las búsquedas sean muy eficientes y en aplicaciones diversas tales como inteligencia artificial o algoritmos de cifrado. Casi todos los sistemas operativos almacenan sus archivos en árboles o estructuras similares a árboles. Además de las aplicaciones citadas, los árboles se utilizan en diseño de compiladores, procesado de texto y algoritmos de búsqueda.

Un árbol consta de un conjunto finito de elementos, denominados **nodos** y de un conjunto finito de líneas dirigidas, denominadas **ramas**, que conectan los nodos.

El número de ramas asociado con un nodo es el GRADO DEL NODO.



Este árbol:

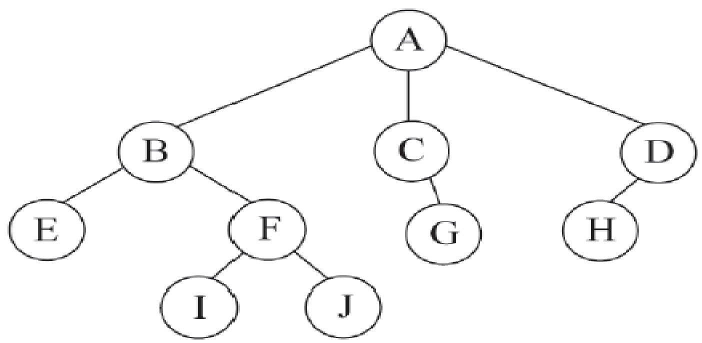
Tiene 9 nodos.

Tiene 8 ramas: AB, AE, AF, BC, BD, FG, FH, FI

Figura 1

Si un árbol no está vacío, entonces el primer nodo se llama raíz. Además del nodo raíz, existen muchos términos utilizados en la descripción de los atributos de un árbol.

En la Figura 2, el nodo A es el NODO RAÍZ. Utilizando el concepto de árboles genealógicos, un nodo puede ser considerado como padre si tiene nodos sucesores. Estos nodos sucesores se llaman hijos.



Por ejemplo:

El nodo B es el padre de los hijos E y F.

El padre de H es el nodo D.

Figura 2.

Un árbol puede representar diversas generaciones en la familia. Los hijos de un nodo y los hijos de estos hijos se llaman descendientes, y el padre y los abuelos de un nodo son sus ascendientes.

Por ejemplo, los nodos E, F, I y J son descendientes de B.

Cada nodo NO raíz tiene un único padre y cada padre tiene cero o más nodos hijos. Dos o más nodos con el mismo padre se llaman hermanos. Un nodo sin hijos, como E, I, J, G y H se llama nodo hoja.

El nivel de un nodo es su distancia al nodo raíz.

La raíz tiene una distancia cero de sí misma, por ello se dice que está en el nivel 0.

Los hijos del nodo raíz están en el nivel 1, sus hijos están en el nivel 2, y así sucesivamente.

Una cosa importante que se aprecia entre los niveles de nodos es la relación entre niveles y hermanos. Los hermanos están siempre al mismo nivel, pero no todos los nodos de un mismo nivel son necesariamente hermanos.

Por ejemplo, en el nivel 2 (Figura 3), C y D son hermanos, al igual que lo son G, H, e I, pero D y G no son hermanos, ya que ellos tienen diferentes padres.

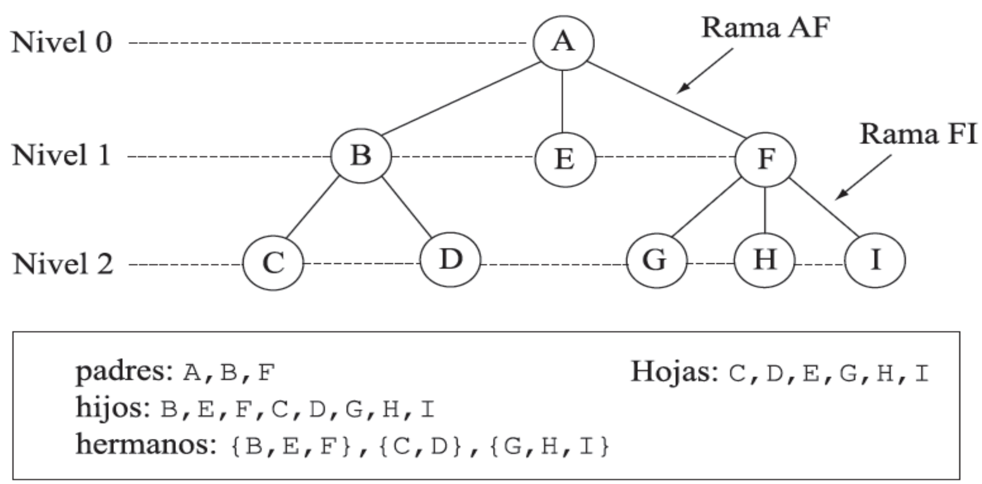


Figura 3.

Un **camino** es una secuencia de nodos en los que cada nodo es adyacente al siguiente. Cada nodo del árbol puede ser alcanzado (se llega a él) siguiendo un único camino que comienza en el nodo raíz.
En la Figura 3, el camino desde el NODO RAÍZ a la hoja I, se representa por AFI. Incluye dos ramas distintas AF y FI.

La altura o profundidad de un árbol es el nivel de la hoja del camino más largo desde la raíz más uno. Por definición , la altura de un árbol vacío es 0.
La Figura 3 contiene nodos en tres niveles: 0, 1 y 2. Su altura es 3.

Un árbol se divide en subárboles. Un subárbol es cualquier estructura conectada por debajo del nodo raíz. Cada nodo de un árbol es la raíz de un subárbol que se define por el nodo y todos sus descendientes. El primer nodo de un subárbol se conoce como el nodo raíz del subárbol y se utiliza para nombrar el subárbol. Además, los subárboles se pueden subdividir en subárboles

En la Figura 3, BCD es un subárbol al igual que E y FGHI. Obsérvese que, por esta definición, un nodo simple es un subárbol. Por consiguiente, el subárbol B se puede dividir en subárboles C y D mientras que el subárbol F contiene los subárboles G, H e I. Se dice que G, H, I, C y D son subárboles sin descendientes.

2.1. ARBOLES BINARIOS

2.1.0. Conceptos fundamentales. Un árbol binario es un **árbol cuyos nodos tienen como máximo dos (2) subárboles**. En un árbol binario, cada nodo puede tener cero, uno o dos hijos (subárboles). Se conoce el nodo de la izquierda como hijo izquierdo y el nodo de la derecha como hijo derecho.

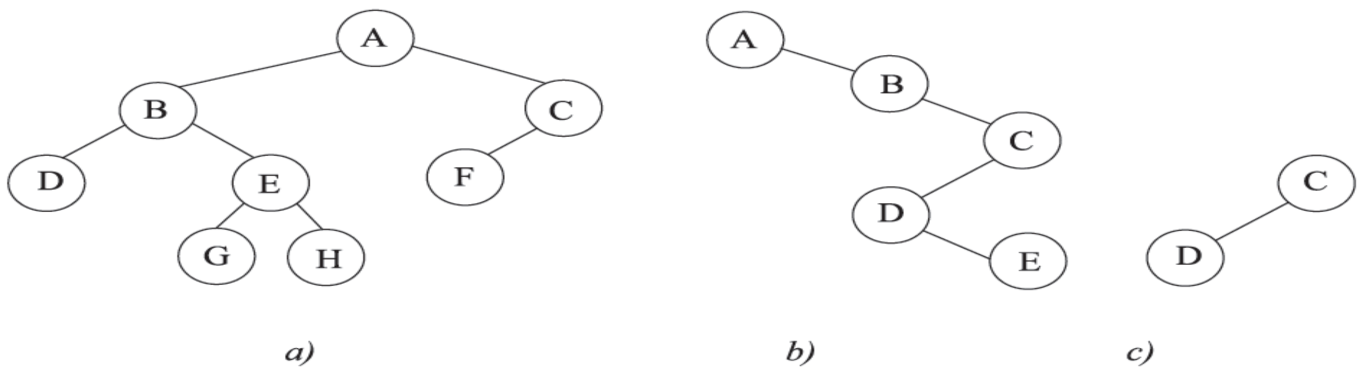


Figura 4.

Un árbol binario es una **estructura recursiva**. Cada nodo es la raíz de su propio subárbol y tiene hijos, que son raíces de árboles, llamados subárboles derecho e izquierdo del nodo, respectivamente.

Un árbol binario se divide en tres subconjuntos disjuntos : *(Disjunto: Que no tiene ningún elemento en común con otro conjunto).*

{R} Nodo raíz.

{I1, I2, ...In} Subárbol izquierdo de R.

{D1, D2, ...Dn} Subárbol derecho de R.

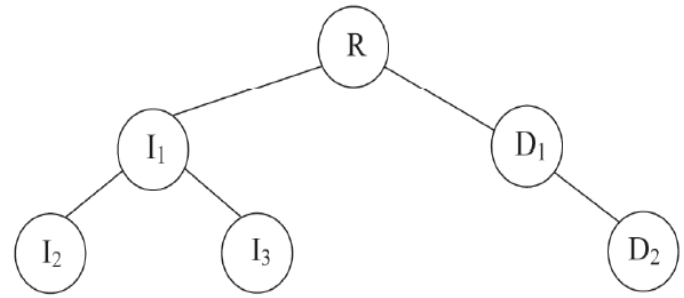


Figura 5.

En cualquier nivel n, un árbol binario puede contener desde 1 hasta 2^n nodos.

El número de nodos por nivel contribuye a la densidad del árbol.

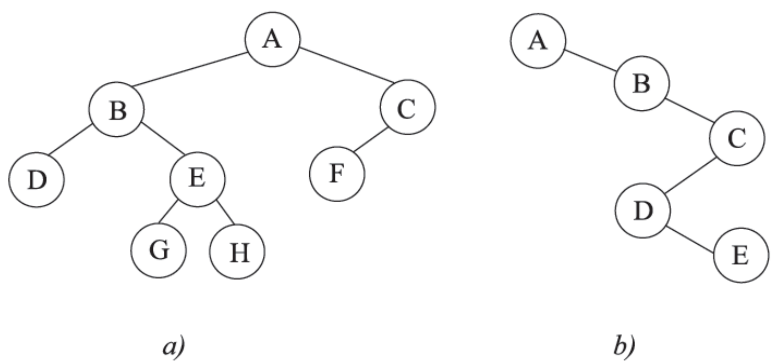


Figura 6.

En la Figura 6(a):

El árbol contiene 8 nodos y profundidad 4.

En la Figura 6(b):

El árbol contiene 5 nodos y profundidad 5.

2.1.1. Equilibrio. La distancia de un nodo a la raíz determina la eficiencia con la que puede ser localizado. Por ejemplo, dado cualquier nodo de un árbol, a sus hijos se puede acceder siguiendo sólo un camino. Esto conduce a una característica muy importante de un árbol binario, su balance o equilibrio. Para determinar si un árbol está equilibrado, se calcula su factor de equilibrio.

El factor de equilibrio de un árbol binario es la diferencia en altura entre los subárboles derecho e izquierdo. Si la altura del subárbol izquierdo es h_I y la altura del subárbol derecho h_D, entonces el factor de equilibrio del árbol B se determina por la siguiente fórmula:

$B = h_D - h_I$	Utilizando esta fórmula, el equilibrio del nodo raíz los árboles de la Figura 6 son (a) -1 y (b) 4.

Un árbol está **perfectamente equilibrado** si su equilibrio o balance es **cero y sus subárboles** son también **perfectamente equilibrados**. Dado que esta definición ocurre raramente se aplica una definición alternativa: un árbol binario está equilibrado si la altura de sus subárboles **difiere en no más de uno y sus subárboles son también equilibrados**; por consiguiente, el factor de equilibrio de cada nodo puede tomar los valores -1, 0, +1.

2.1.2. Árboles binarios completos. Un árbol binario completo de profundidad **n** es un árbol en el que para cada nivel, del 0 al nivel n-1, tiene un conjunto lleno de nodos, y todos los nodos hoja a nivel **n** ocupan las posiciones **más a la izquierda** del árbol. Un árbol binario completo es un árbol binario en el que cada nodo tiene 0 ó 2 hijos.

Un árbol binario completo que contiene 2^n nodos a nivel **n** es un árbol lleno.
Un árbol lleno es un árbol binario que tiene el máximo número de entradas para su altura. Esto sucede cuando el último nivel está lleno. La Figura 7 muestra un árbol binario completo; y la Figura 8 se corresponde con uno lleno.

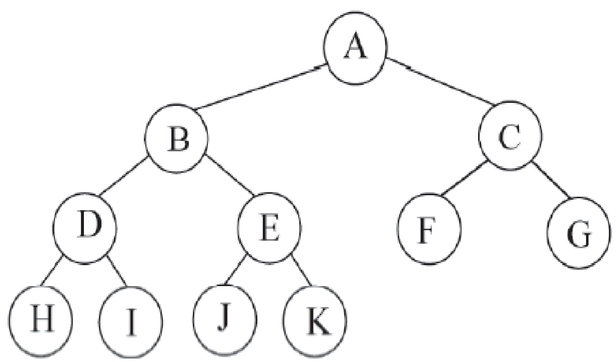


Figura 7.

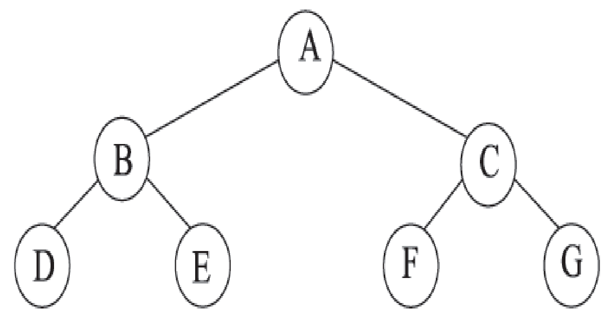


Figura 8.

El último caso de árbol es un tipo especial, denominado árbol degenerado (Figura 9), en el que hay un solo nodo hoja (E) y cada nodo no hoja sólo tiene un hijo. Un árbol degenerado es equivalente a una lista enlazada.

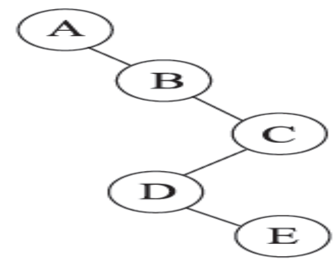


Figura 9.

El número de nodos, n, en un árbol binario completo de profundidad k+1 (Niveles 0 a k) cumple la desigualdad:

$$2^k \leq n < 2^{(k+1)}$$

El número de nodos en un árbol lleno, se corresponde con $n = 2^{(k+1)} - 1$

Por ejemplo, un árbol lleno de profundidad 4 (niveles 0 a 3) tiene $2^4 - 1 = 15$ **nodos.**

Aplicando logaritmos a la ecuación con desigualdad anterior: $k \leq \log_2 (n) < (k + 1)$

De allí se deduce que la altura o profundidad (h) de un árbol binario completo de n nodos es:

$$h = | \log_2 (n) | + 1$$

Ejemplos:

<u>(1) Calcular la profundidad máxima y mínima de un árbol con 5 nodos.</u>

<table>
<tr><td colspan="2">La profundidad máxima de un árbol con 5 nodos es 5, se corresponde con un árbol degenerado (como el de la Figura 9).</td></tr>
<tr><td>La profundidad mínima h (número de niveles más uno) de un árbol con 5 nodos, aplicando la inecuación del número de nodos de un árbol binario completo es:</td><td>$k \leq \log_2 (5) < (k + 1)$
$h = | \log_2 (5) | + 1$
$h = | 2.32 | + 1$
$h = 2 + 1$

La profundidad mínima es $h = 3$</td></tr>
</table>

<u>(2) Suponiendo que se tiene n = 10.000 elementos que van a ser los nodos de un árbol binario completo. Determinar la profundidad del árbol.</u>

En el árbol binario completo con n nodos, la profundidad del árbol es $h = | \log_2 (n) | + 1$, que también se denomina la distancia del camino más largo desde la raíz a un nodo más uno.

$$h = | \log_2 (10000) | + 1 = | 13.28 | + 1 = 14$$

2.1.3. Primer acercamiento a los recorridos de un árbol. Para visualizar o consultar los datos almacenados en un árbol se necesita recorrer el árbol o visitar los nodos del mismo. Al contrario que las listas enlazadas, los árboles binarios no tienen realmente ~~un primer valor~~, un segundo valor, un tercer valor, etc. Se puede afirmar que el nodo raíz viene el primero, pero, ¿quién viene a continuación? Existen diferentes métodos, basados en dos (2) enfoques: recorrido en profundidad y recorrido en anchura.

El recorrido de un árbol binario requiere que cada nodo sea procesado (visitado) una y sólo una vez, en una secuencia predeterminada. Dado un árbol binario que consta de raíz, un subárbol izquierdo y un subárbol derecho, se pueden definir tres tipos de secuencia de ***recorrido en profundidad***:

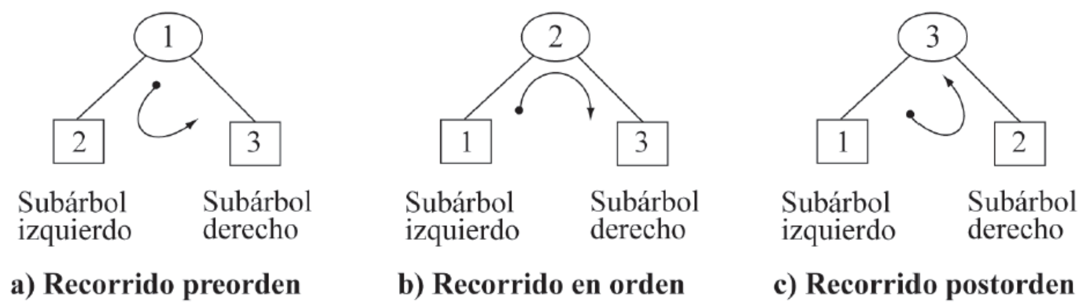


Figura 10.

La designación tradicional de los recorridos utiliza: para el nodo raíz (N), para el subárbol izquierdo (I) y para el subárbol derecho (D).

<u>Recorrido PREORDEN (NID):</u> el **nodo raíz va antes** que los subárboles:

1. Visitar el nodo raíz (N).

2. Recorrer subárbol izquierdo (I) en PREORDEN.

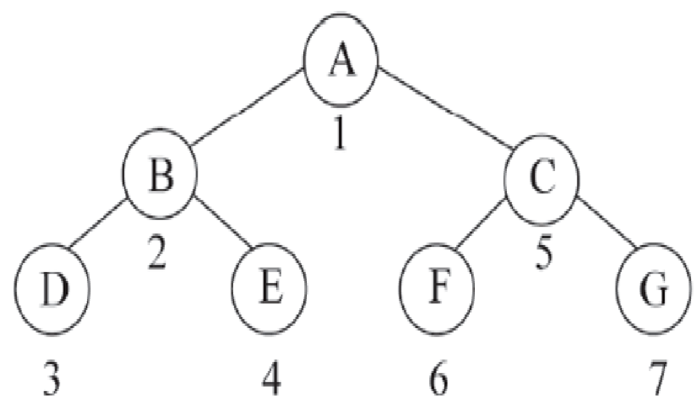
3. Recorrer subárbol derecho (D) en PREORDEN.

Si utilizamos el recorrido PREORDEN del árbol de la Figura 11,

Se visita **primero la raíz** (nodo A).

A continuación, se visita el **subárbol izquierdo de A**, que consta de los nodos B, D y E. Dado que el subárbol es a su vez un árbol, se visitan los nodos utilizando el mismo orden (NID). Por consiguiente, se visita primero el nodo B, después D (izquierdo) y por último E (derecho).

A continuación, se visita el **subárbol derecho de A**, que es un árbol que contiene los nodos C, F y G. De nuevo, siguiendo el mismo orden (NID), se visita primero el nodo C, a continuación F (izquierdo) y, por último, G (derecho).



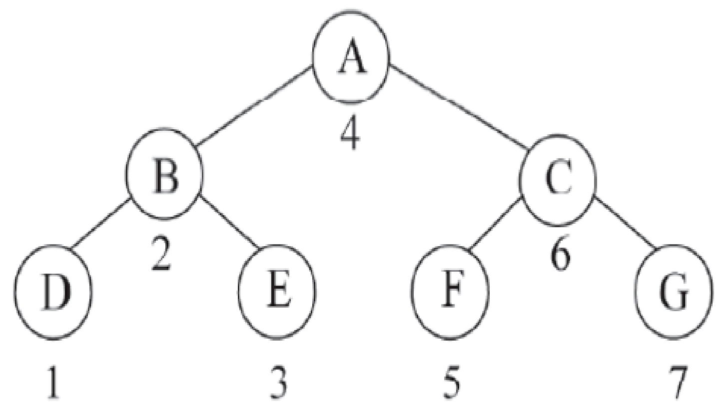
RECORRIDO: A , B , D , E , C , F , G

Figura 11.

<u>Recorrido EN ORDEN (IND):</u> También llamado "inorder" procesa primero el subárbol izquierdo, después el raíz y, a continuación, el subárbol derecho. El **nodo raíz se procesa entre** los subárboles.

1. Recorrer el subárbol izquierdo (I) EN ORDEN.
2. Visitar el nodo raíz (N).
3. Recorrer el subárbol derecho (D) EN ORDEN.

Si utilizamos el recorrido INORDEN del árbol de la Figura 12,

El **primer subárbol recorrido es el subárbol izquierdo** del nodo raíz (árbol cuyo nodo contiene la letra B). Este subárbol es, a su vez, otro árbol con el nodo B como raíz, por lo que siguiendo el orden IND, se visita primero D, a continuación B (nodo raíz) y, por último, E (derecha). **Después, se visita el nodo raíz, A.** Por último **se visita el subárbol derecho de A**, siguiendo el orden IND se visita primero F, después C (nodo raíz) y por último G.	 **RECORRIDO: D , B , E , A , F , C , G** Figura 12.

<u>Recorrido en POSTORDEN (IDN):</u> procesa el nodo raíz (post) **después de que los subárboles izquierdo y derecho se hayan procesado**. Comienza situándose en la hoja más a la izquierda y se procesa. A continuación, se procesa su subárbol derecho. Por último, se procesa el nodo raíz.

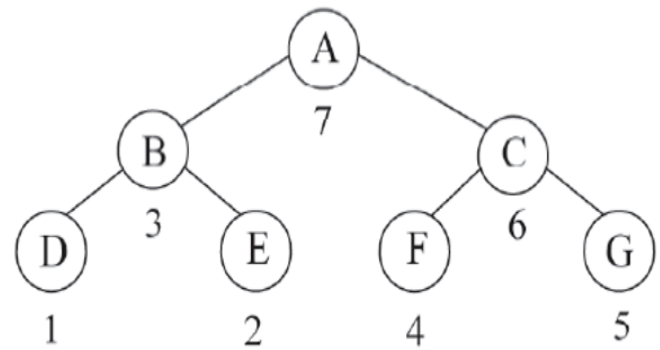
1. Recorrer el subárbol izquierdo (I) en POSTORDEN.
2. Recorrer el subárbol derecho (D) en POSTORDEN.
3. Visitar el nodo raíz (N).

Si utilizamos el recorrido POSTORDEN del árbol de la Figura 13,

Se visita **primero el subárbol izquierdo de A**. Este subárbol consta de los nodos B, D y E, y siguiendo el orden IDN, se visitará primero D (izquierdo), luego E (derecho) y, por último, B (nodo).

A continuación, se visita el subárbol derecho de A que consta de los nodos C, F y G. Siguiendo el orden IDN para este árbol, se visita primero F (izquierdo), después G (derecho) y, por último, C (nodo).

Finalmente se visita el nodo raíz, A.



RECORRIDO: D , E , B , F , G , C , A

Figura 13.

--- *FIN DE SECCIÓN*