

Entendemos por **árbol** una estructura de datos formada por varios objetos colocados en una forma semejante a un árbol biológico, es decir, los objetos parece que están sobre las ramas de un árbol, recordemos un árbol genealógico, uno de los objetos es la raíz, y los demás están colocados en las ramas. Así, un árbol está formado por varios objetos, (un número finito), y varias líneas llamadas ramas, que sirven para unir o conectar los objetos.

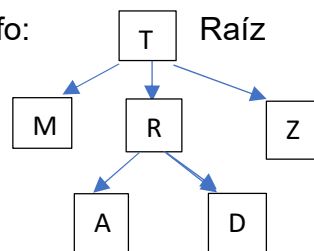
“Se define un árbol como una estructura de datos formada por varios objetos, (una cantidad finita), llamados nodos y varias líneas que conectan esos nodos, (la cantidad de líneas es finita), denominadas ramas”. (Joyanes Aguilar & Zahonero Martínez, 2008)

“En modo formal un árbol de tipo T, se define como la estructura homogénea que resulta de la concatenación de un elemento tipo T (la raíz), y de un número finito de árboles ajenos o disjuntos, denominados subárboles; un árbol puede ser una estructura vacía”. (Silva Guardati, 2006)

Los subárboles son árboles también, así un árbol es una estructura recursiva, y por tanto, es posible utilizar rutinas recursivas para generar un árbol.

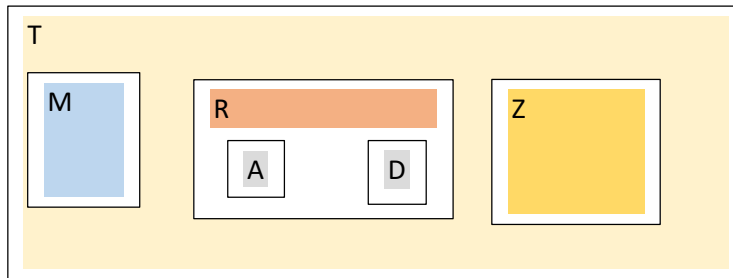
Existen varias áreas de conocimiento donde se han utilizado árboles, por ejemplo, en los árboles genealógicos, en los circuitos eléctricos, y algunas otras, hay diversas formas de representar un árbol, entre ellas se tiene:

1. Con un grafo:



Nota: Esta representación es la más usual, su uso ha introducido el término árbol dado su parecido muy similar con un vegetal árbol: raíz, ramas, hojas. En esta representación dibujamos la raíz arriba, y el árbol en sí hacia abajo.

2. Usando diagramas de VENN



3. usando la notación decimal de Dewey

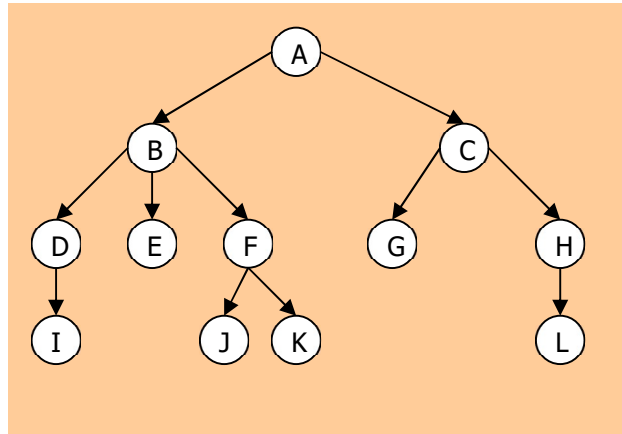
1. T, 1.1 M, 1.2 R, 1.2.1 A, 1.2.2 D, 1.3 Z

Características de los árboles

Las más importantes son:

- a) Cualquier árbol no vacío tiene un nodo raíz que es único.
- b) Se dice que un nodo X es descendiente directo de un nodo Y, si el nodo Y apunta al nodo X, es muy común decir que **X es hijo de Y**.
- c) Decimos que un nodo X es antecesor directo de un nodo Y, cuando el nodo X apunta al nodo Y, es común decir que **X es el padre de Y**.
- d) Decimos que todos los nodos descendientes directos o hijos de un mismo nodo padre, son **hermanos**.
- e) Si nodo no tiene hijos o ramificaciones, es una **hoja**.
- f) Cualquier nodo que no es raíz, ni es hoja, es nodo **interior**
- g) El **Grado de un nodo** es el número de hijos de un nodo. El **Grado de un árbol** es el grado máximo de cada uno de sus nodos.
- h) El **Nivel** consta del número de líneas, enlaces o arcos que se deben recorrer para llegar a un nodo en particular. La raíz tiene nivel 1 por definición.
- i) La **Altura** del árbol es el máximo nivel de cada uno de los nodos del árbol.

Ejemplo: Consideremos el árbol mostrado en la figura de abajo:



Se tiene:

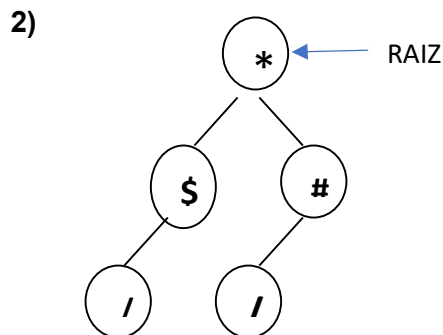
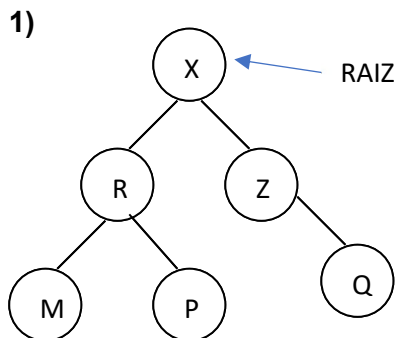
- A es raíz del árbol; B es hijo de A; C es hijo de A; D es hijo de B; E es hijo de B.
- A es padre de B; B es padre de D; D es padre de I; C es padre de G.
- B y C son hermanos; D, E y F son hermanos; G y H son hermanos.
- I, E, J, K, G y L son nodos hoja; B, D, F, C y H son nodos interiores.
- El grado del nodo A es 2; El grado del nodo B es 3; El grado del nodo C es 2.
- El grado del árbol es 3.
- El nivel del nodo A es 1; El nivel del nodo B es 2; El nivel del nodo D es 3.
- La altura del árbol es 4.

Árboles Binarios

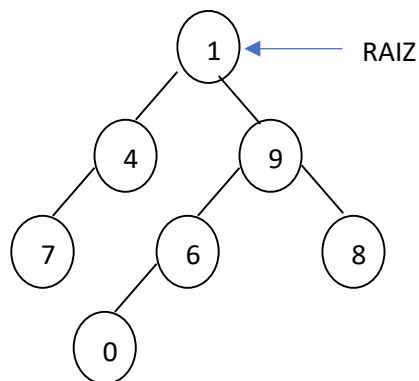
“Un **árbol binario** es un árbol donde cada nodo no puede tener más de dos subárboles, es decir, cada nodo puede tener uno, dos, o cero hijos o subárboles”.
(Joyanes Aguilar & Zahonero Martínez, 2008)

El nodo de la izquierda se dice hijo izquierdo, el nodo de la derecha se dice hijo derecho

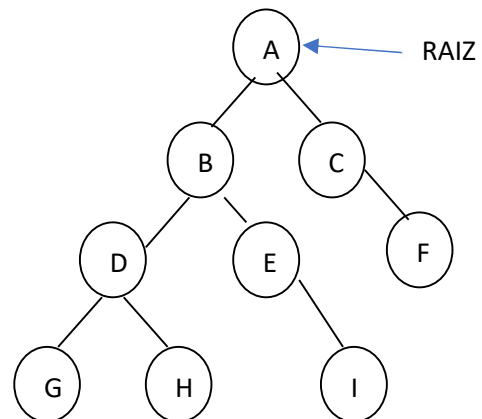
Ejemplos:



3)



4)



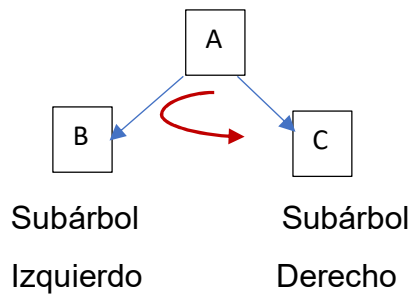
Algunas características de los árboles binarios son:

- Si se tiene que N1 es la raíz de un árbol binario y que N2 es la raíz del subárbol izquierdo o del subárbol derecho; se suele decir que N1 es padre de N2 y que N2 es hijo izquierdo o hijo derecho de N1.
- Se le llama nodo hoja a un nodo que no tiene nodos hijos.
- Decimos que un nodo N1 es nodo ancestro de N2, decimos que N2 es un nodo descendiente de N1, si N1 es nodo padre del nodo N2 o es nodo padre de algún ancestro de N2.
- Se tiene que un nodo N2 es descendiente izquierdo o descendiente derecho del nodo N1, si N2 es hijo izquierdo o hijo derecho de N1.
- Dos nodos se dicen hermanos, si son hijo izquierdo e hijo derecho del mismo nodo padre.
- El nivel de la raíz del árbol es cero, el nivel de cualquier otro nodo en el árbol binario es mayor en una unidad que el nivel de su nodo padre.
- Un árbol binario completo de nivel N es tal que, cada nodo de nivel N es un nodo hoja, y cualquier nodo de nivel menor que N no tiene subárboles izquierdos y derechos vacíos. (Silva Guardati, 2006)

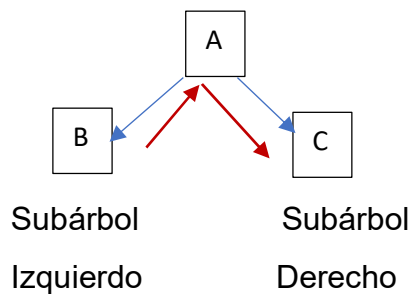
Consultas en los nodos de un árbol binario

Supongamos que tenemos un árbol binario y deseamos saber que datos hay en cada uno de sus nodos, es decir, deseamos consultar, visitar, o recorrer cada uno de sus nodos para leer, modificar, o extraer la información contenida. Es común llamar al hecho de visitar los nodos de un árbol como el “recorrido del árbol, dicho recorrido supone visitar una sola vez cada nodo del árbol”. (Joyanes Aguilar & Zahonero Martínez, 2008)

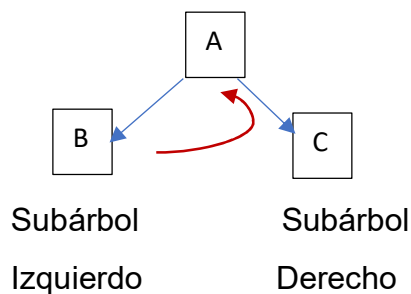
Dado que los nodos de un árbol binario no tienen un orden natural, es posible utilizar varias formas para el recorrido. Supongamos que tenemos un árbol binario con su raíz, su subárbol izquierdo y su subárbol derecho, se pueden establecer tres tipos de recorrido en profundidad, la forma de cada recorrido se muestra en las imágenes que siguen:



a) El recorrido inicia en la raíz A, sigue el subárbol izquierdo al terminar, sigue el subárbol derecho. Este recorrido es llamado **pre orden**.



b) El recorrido inicia en el subárbol izquierdo al terminar, sigue con la raíz, y después sigue el subárbol derecho. Este recorrido es llamado **en orden**.



c) El recorrido inicia en el subárbol izquierdo al terminar, sigue el subárbol derecho al terminar, sigue la raíz. Este recorrido es llamado post orden.

Por tanto, se tienen tres modos de recorrido de un árbol binario en forma recursiva, de modo tal que, recorrer un árbol binario consiste en visitar la raíz, y recorrer los subárboles izquierdo y derecho; esas formas de recorrer el árbol son diferentes en el orden en que visitan los nodos del árbol.

Así se pueden definir tres formas de recorrer un árbol binario:

1. Recorrer un árbol binario no vacío en **PREORDEN**, consiste en: Visitar la raíz; Recorrer el subárbol de la izquierda en PREORDEN; Recorrer el subárbol de la derecha en PREORDEN.
2. Recorrer un árbol binario no vacío en **ENTRE ORDEN**, consiste en: Recorrer el subárbol de la izquierda en entre orden; Visitar la raíz; Recorrer el subárbol de la derecha entre orden.
3. Recorrer un árbol binario no vacío en **POS ORDEN**, consiste en: Recorrer el subárbol de la izquierda en POS orden; Recorrer el subárbol de la derecha en POS orden; Visitar la raíz.(Cairó, Guardati. 2006. Pág.198).

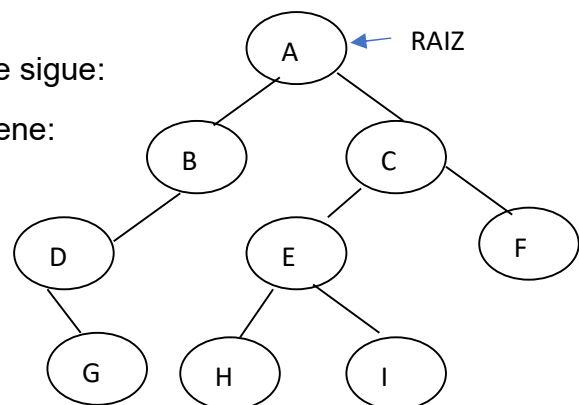
Ejemplos: Consideremos el árbol binario que sigue:

Al aplicar los recorridos al árbol binario se tiene:

PRE ORDEN: A, B, D, G, C, E, H, I, F

ENTRE ORDEN: D, G, B, A, H, E, I, C, F

POS ORDEN: G, D, B, H, I, E, F, C, A



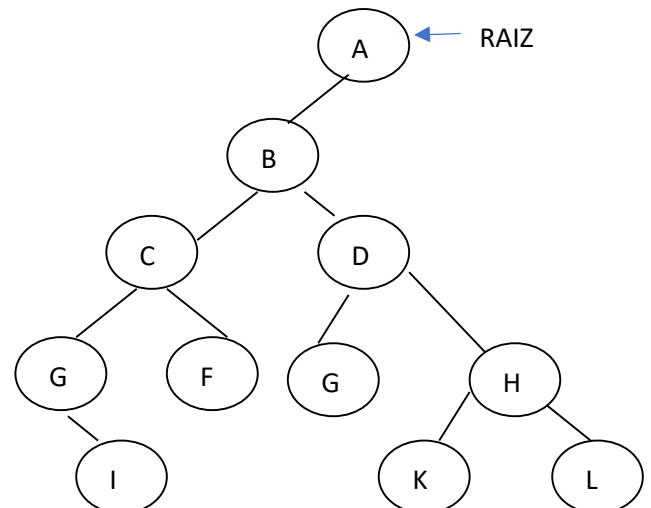
Consideremos el árbol binario que sigue:

Los diferentes recorridos del árbol binario son:

PRE ORDEN: A, B, C, E, I, F, J, D, G, H, K, L

ENTRE ORDEN: E, I, C, F, J, B, G, D, K, H, L, A

POS ORDEN: I, E, J, F, C, G, K, L, H, D, B, A



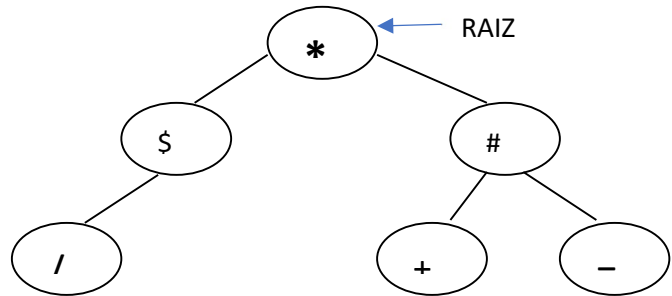
Consideremos el árbol binario que sigue:

Los recorridos del árbol binario son:

PRE ORDEN: *, \$, /, #, +, =

ENTRE ORDEN: /, \$, *, +, #, =

POS ORDEN: /, \$, +, =, #, *



Referencias

Joyanes Aguilar, L., & Zahonero Martínez, I. (2008). *Estructura de datos en Java*. Madrid, España: Mc Graw Hill Interamericana.

Silva Guardati, O. (2006). *Estructuras de Datos*. México: McGraw Hill Interamericana.