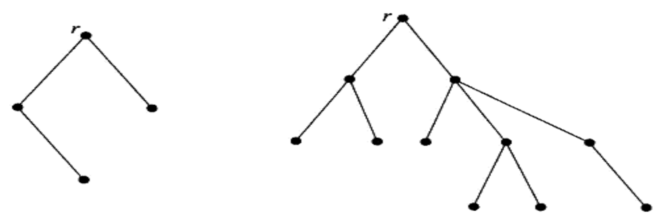
Sección 5.2 Árboles y sus Representaciones. Pág. 371

Terminología de Árboles

Un tipo especial de grafo es llamado **árbol** y son muy útiles para la representación de datos.

Definición: Árbol. Un árbol es un grafo conectado y acíclico, con un nodo designado como raíz (root) del árbol.



(a) (b)

FIGURE 5.31

Como un árbol es un grafo conectado, hay una trayectoria desde la raíz a cualquier otro nodo en el árbol, y como el árbol es acíclico la trayectoria es única.
La profundidad de un nodo en un árbol, es la longitud de la trayectoria de root al nodo.
La profundidad (altura) de un árbol es la máxima profundidad de cualquier nodo en el árbol.
Un nodo sin hijos es llamado **hojas del árbol**.
Un bosque es un grafo acíclico (no necesariamente conectado), además **un bosque** es una colección disyunta de árboles.
Árboles binarios, donde cada nodo tiene a lo más dos hijos, son de gran interés particular. En un árbol binario, cada hijo de un nodo se designa como hijo izquierdo ó hijo derecho.
Un árbol binario lleno ocurre cuando todos los nodos internos tienen dos hijos y todas las hojas tienen la misma profundidad. La figura 5.33 muestra un árbol binario de altura 4, y la figura 5.34 muestra un árbol binario lleno de altura 3.
Un árbol binario completo es casi un árbol binario lleno, pero no esta completamente lleno de hojas. La figura 5.35 muestra un árbol binario completo de altura 3.

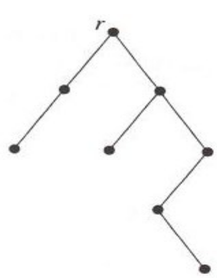


FIGURE 5.33

FIGURE 5.34

FIGURE 5.35

Práctica 18. Pag. 372 Responda las siguientes preguntas acerca del árbol binario de la siguiente figura. (Asuma que el nodo 1 es la raíz del árbol.)

a. ¿Cuál es la altura?

b. ¿Cuál es el hijo izquierdo del nodo 2?
c. ¿Cuál es la profundidad del nodo 5?

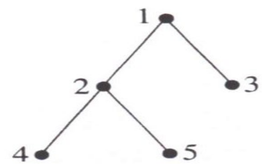


FIGURE 5.36

Aplicaciones de árboles

Los árboles se usan para representar la estructura de cómo están distribuidos los archivos en un computador. También para representar operaciones algebraicas.

Ejemplo 22. Pág. 373 Un virus de computador se esparce vía email. Cada segundo, 4 nuevas máquinas son infectadas. 4^n máquinas han sido infectadas después de n segundos

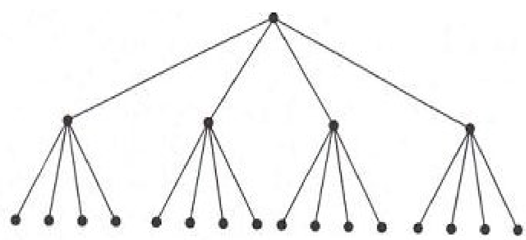


FIGURE 5.39

Ejemplo 23. Pág. 374 Expresiones algebraicas que involucran operaciones binarias pueden ser representadas por árboles binarios etiquetados.

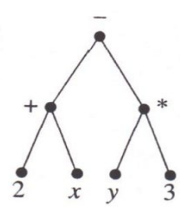


FIGURE 5.40

Práctica 19. Pág. 374 Cuál es el árbol que representa la expresión algebraica (2 + 3) * 5 ?

Representación de Árboles Binarios

Como un árbol es un grafo, las representaciones para grafos en general discutidas en la sección 5.1, también pueden ser usadas para árboles. Árboles binarios, sin embargo, tienen características especiales que se deben mostrar en la representación, como son el hijo derecho e hijo izquierdo. El equivalente de una matriz de adyacencia es un arreglo de dos columnas, donde los datos de cada nodo son el hijo izquierdo y el hijo derecho de cada nodo. La equivalente a una lista de adyacencia es una colección de registros con tres campos, conteniendo respectivamente, el nodo actual, un puntero para el nodo hijo izquierdo y otro para el nodo hijo derecho.

Ejemplo 24. Pág. 374 Para el árbol de la siguiente figura, se muestran sus posibles representaciones.

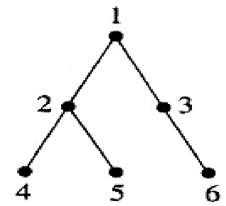
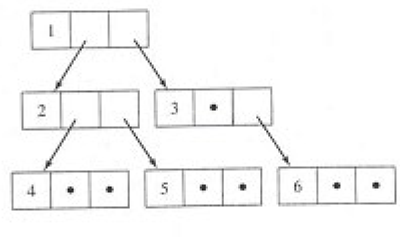


FIGURE 5.41

	Left child	Right child
1	2	3
2	4	5
3	0	6
4	0	0
5	0	0
6	0	0

(a)



(b)

FIGURE 5.42

Sección 5.2 Algoritmos de Recorrido de Árboles. Pág. 375

Si una estructura de árbol es usada para almacenar datos, esta es frecuentemente de gran ayuda para tener un mecanismo sistemático para mostrar los valores de los datos almacenados en todos los nodos. Esto puede ser realizado recorriendo el árbol, que es, visitando cada nodo en la estructura del árbol. Los tres algoritmos recursivos más comunes para recorrer árboles son preorder, inorder y postorder.(preorden, intermedio, orden final respectivamente)

1. Recorrido Preorden.

El root del árbol es visitado primero y luego los subárboles son procesados de izquierda a derecha, cada uno en un recorrido preorden. **(root, izquierda, derecha)**

ALGORITMO PREORDER
Preorder(tree T)
//Escriba los nodos de un árbol con raíz r en preorder.

Write (r)
for i = 1 to t **do**
Preorder(T_i)
end for
end Preorder

2. Recorrido Intermedio

El árbol izquierdo es procesado por un recorrido intermedio, luego root es visitado, y luego los siguientes árboles son procesados de izquierda a derecha, cada uno en un recorrido intermedio. **(Izquierda, root, derecha)**

ALGORITMO INORDER
Inorder(tree T)
//Escriba los nodos de un árbol con raíz r en inorder.

Inorder (T_1)
Write (r)
for i = 2 to t **do**
Inorder(T_i)
end for
end Inorder

3. Recorrido Orden Final

El root es visitado de último, después que todos los árboles han sido procesados de izquierda a derecha, cada uno en recorrido orden final. **(Izquierda, derecha, root)**

ALGORITMO POSTORDER
Postorder(tree T)
//Escriba los nodos de un árbol con raíz r en postorder.
for i = 1 to t **do**
Postorder(T_i)
end for
write (r)
end Postorder

Ejemplos 25 y 26. Pág. 377 Para el siguiente árbol, escriba los recorridos para cada uno de los métodos.

i) Preorder. a, b, d, e, c, f, h, i, g
ii) Inorder. d, b, e, a, h, f, i, c, g
iii) Postorder. d, e, b, h, I, f, g, c, a

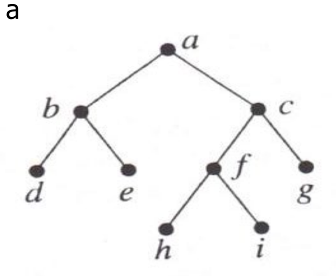


FIGURE 5.45

Ejemplo 27. Pág. 377 Para el siguiente árbol, escriba los recorridos para cada uno de los métodos.

i) Preorder. a, b, d, i, e, f, c, g, j, k, h
ii) Inorder. i, d, b, e, f, a, j, g, k, c, h

Universidad AUTÓNOMA de Occidente

iii) Postorder. i, d, e, f, b, j, k, g, h, c, a

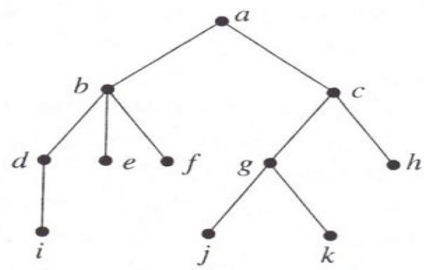


FIGURE 5.46

Práctica 21. Pág. 378 Para el siguiente árbol, escriba los recorridos para cada uno de los métodos.

i) Preorder. _, _, _, _, _, _, _, _, _,
ii) Inorder. _, _, _, _, _, _, _, _, _,
iii) Postorder. _, _, _, _, _, _, _, _, _,

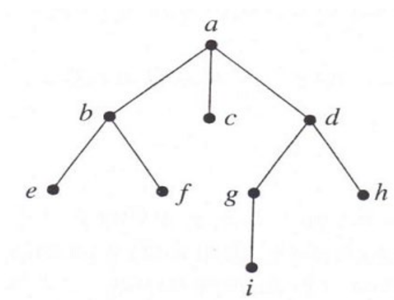
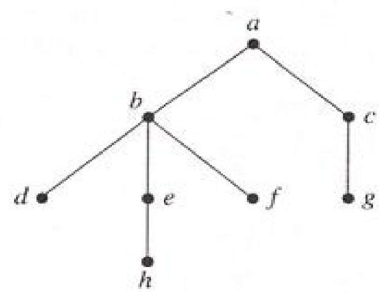


FIGURE 5.47

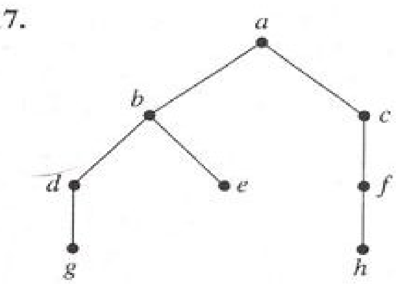
EJERCICIOS 5.2 Pág. 381

Para los ejercicios del 16 al 21, escriba la lista de nodos como resultados de los recorridos preorder, inorder y postorder de cada árbol.

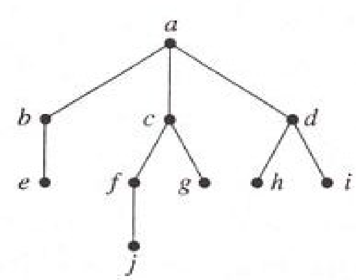
★16.



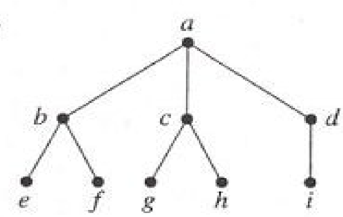
17.



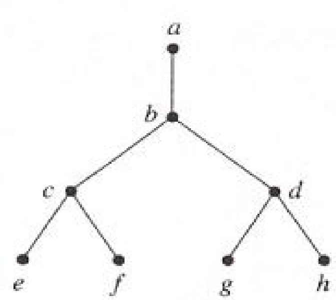
18.



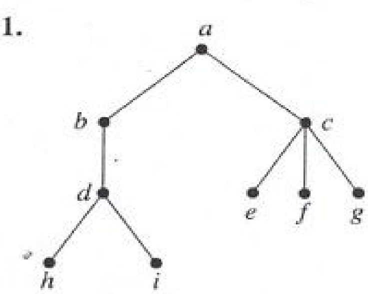
19.



★20.



21.



Árboles Binarios que representan expresiones aritméticas

Ejemplo 28. Pág. 379 El ejemplo 23 presentado en una clase anterior, muestra como una expresión algebraica puede ser representada por un árbol binario. Si hacemos un recorrido inorder del árbol, tenemos la expresión algebraica original. Para el árbol presentado abajo, por ejemplo, un recorrido inorder da la expresión

$$(2 + x) * 4$$

donde los paréntesis son adicionados cuando se completa el recorrido de un subárbol.

Esta forma de expresión algebraica, donde los símbolos aparecen entre dos operandos, es llamada **Notación Infija ó Infix**. Los paréntesis son necesarios para indicar el orden de las operaciones. Sin los paréntesis la expresión sería 2 + x * 4, la cual también es una expresión infija, pero como el orden de precedencia de la multiplicación es mayor sobre la adición, esta no expresa lo que realmente se quiere.

Universidad AUTÓNOMA de Occidente

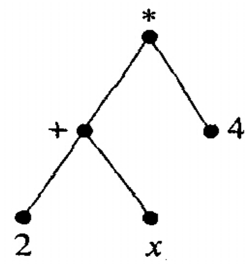


FIGURE 5.48

Un recorrido preorder de este árbol da la expresión

$$* + 2 \times 4$$

aquí los símbolos de la operación preceden los operandos. Esta forma de expresión se denomina **Notación Prefija ó Notación Polaca.** La expresión puede ser traducida a infija así:

$$* + 2\ x\ 4 \rightarrow * (2 + x)\ 4 \rightarrow (2 + x) * 4$$

Un recorrido postorder da la expresión

$$2\ x + 4\ *$$

donde los símbolos de operación están después de los operandos. Esta forma de expresión se denomina **Notación Postfija ó Notación Polaca Inversa (reverse) ó RPN.** La expresión puede ser traducida a infija así:

$$2\ x + 4\ * \rightarrow (2 + x)\ 4\ * \rightarrow (2 + x) * 4$$

Ninguna de las notaciones prefija y postfija requiere paréntesis para eliminar ambigüedades. Los compiladores cambian las expresiones algebraicas en los programas de computador desde notación infija a postfija para mayor eficiencia de procesamiento.

Práctica 22. Pág. 378 Escriba la expresión de árbol para

$$a + (b * c - d)$$

y escriba la expresión en notaciones prefija y postfija.

EJERCICIOS 5.2 Pág. 381

Para los ejercicios del 3 al 6 dibuje el árbol asociado.

3. $[(x - 2) * 3] + (5 + 4)$

4. $[(2 * x - 3 * y) + 4 * z] + 1$

5. $1 - (2 - [3 - (4 - 5)])$

6. [(6 / 2) * 4] + [(1 + x) * (5 + 3)]

22. Escriba en notaciones prefija y postfija: 3 **/ 4 + (2 – y)**

23. Escriba en notaciones prefija y postfija: (**x * y + 3 / z)** * **4**

24. Escriba en notaciones infija y postfija: **- * + 2 3 * 6 x 7**

25. Escriba en notaciones infija y postfija: - **+ - x y z w**

26. Escriba en notaciones prefija e infija: 4 **7 x - * z +**

27. Escriba en notaciones prefija e infija: **x 2 w + y z * - /**

28. Dibuje árbol el cuyo recorrido preorder es

a, b, c, d, e

y cuyo recorrido inorder es

b, a, d, c, e

29. Dibuje árbol el cuyo recorrido inorder es

f, a, g, b, h, d, i, c, j, e

y cuyo recorrido postorder es

f, g, a, h, i, d, j, e, c, b