

# PROGRAMACIÓN I

LECTURA

## UNIDAD 3 ESTRUCTURAS DINÁMICAS 2

### ARBOLES

Autor de contenidos:  
Nicolás Battaglia



## OBJETIVOS

### ARBOLES BINARIOS

En ciencias de la computación, un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno. Usos comunes de los árboles binarios son los árboles binarios de búsqueda

En teoría de grafos, se usa la siguiente definición: «Un árbol binario es un grafo conexo, acíclico y no dirigido tal que el grado de cada vértice no es mayor a 3». De esta forma solo existe un camino entre un par de nodos.

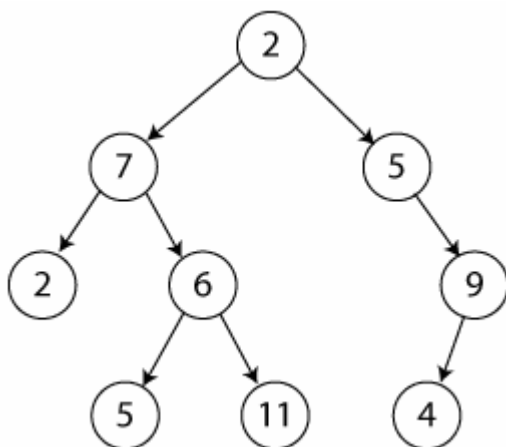
- **Grafo** es una representación gráfica de diversos puntos que se conocen como nodos o vértices, los cuales se encuentran unidos a través de líneas que reciben el nombre de aristas.
- **Conexo** que está conectado o guarda relación entre todas sus partes
- **Acíclico** implica que no tiene ciclos; esto significa que para cada vértice  $v$ , no hay un camino directo que empiece y termine en  $v$ , no tiene sentido que un vértice tenga un camino directo a él mismo
- **No dirigido** implica que las aristas son no dirigidas El grafo no dirigido es aquel que no tiene sentido su arista

Un árbol binario con enraizado es como un grafo que tiene uno de sus vértices, llamado raíz, de grado no mayor a 2. Con la raíz escogida, cada vértice tendrá un único padre, y nunca más de dos hijos. Si rehusamos el requerimiento de la conectividad, permitiendo múltiples componentes conectados en el grafo, llamaremos a esta última estructura un bosque'.

Un árbol binario es un árbol en el que ningún nodo puede tener más de dos 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.

### Recorridos sobre árboles





En base al gráfico anterior, se detallan y ejemplifican las diferentes formas de recorrerlo.

### **Recorridos en profundidad**

El método de este recorrido es tratar de encontrar de la cabecera a la raíz en nodo de unidad binaria. Ahora pasamos a ver la implementación de los distintos recorridos:

#### **Recorrido en preorden**

En este tipo de recorrido se realiza cierta acción (quizás simplemente imprimir por pantalla el valor de la clave de ese nodo) sobre el nodo actual y posteriormente se trata el subárbol izquierdo y cuando se haya concluido, el subárbol derecho. Otra forma para entender el recorrido con este método sería seguir el orden: nodo raíz, nodo izquierda, nodo derecha.

En el árbol de la figura el recorrido en preorden sería: 2, 7, 2, 6, 5, 11, 5, 9 y 4.

#### **Recorrido en postorden**

En este caso se trata primero el subárbol izquierdo, después el derecho y por último el nodo actual. Otra forma para entender el recorrido con este método sería seguir el orden: nodo izquierda, nodo derecha, nodo raíz.

En el árbol de la figura el recorrido en postorden sería: 2, 5, 11, 6, 7, 4, 9, 5 y 2.

#### **Recorrido en inorden**

En este caso se trata primero el subárbol izquierdo, después el nodo actual y por último el subárbol derecho. En un ABB este recorrido daría los valores de clave ordenados de menor a



mayor. Otra forma para entender el recorrido con este método sería seguir el orden: nodo izquierda, nodo raíz, nodo derecha.

En el árbol de la figura el recorrido en inorden sería: 2, 7, 5, 6, 11, 2, 5, 4, 9.

### **Recorridos en amplitud (o por niveles)**

En este caso el recorrido se realiza en orden por los distintos niveles del árbol. Así, se comenzaría tratando el nivel 1, que solo contiene el nodo raíz, seguidamente el nivel 2, el 3 y así sucesivamente.

En el árbol de la figura el recorrido en amplitud sería: 2, 7, 5, 2, 6, 9, 5, 11 y 4.

Al contrario que en los métodos de recorrido en profundidad, el recorrido por niveles no es de naturaleza recursiva. Por ello, se debe utilizar una cola para recordar los subárboles izquierdos y derecho de cada nodo.

El esquema algoritmo para implementar un recorrido por niveles es exactamente el mismo que el utilizado en la versión iterativa del recorrido en preorden pero cambiando la estructura de datos que almacena los nodos por una cola.