un árbol es una estructura de datos jerárquica que consiste en nodos, donde cada nodo tiene un valor y puede tener cero o más nodos hijos, los árboles se utilizan en diversas aplicaciones como en la representación de expresiones, la organización de datos jerárquicos y las estructuras de datos avanzadas como los árboles de búsqueda

**Terminología de Árboles**

raíz: el nodo superior de un árbol

nodo: un elemento del árbol

hijo: un nodo descendente de otro nodo

padre: un nodo con un enlace directo a uno o más nodos hijos

hoja: un nodo que no tiene hijos

profundidad: la longitud del camino desde la raíz hasta un nodo

altura: la longitud del camino desde un nodo hasta la hoja más lejana

**Tipos de árboles**

árbol binario: un árbol en el que cada nodo tiene como máximo dos hijos

árbol binario de búsqueda: un árbol binario donde para cada nodo todos los valores del subárbol izquierdo son menores y todos los valores del subárbol derecho son mayores

árbol avl: un árbol binario de búsqueda auto balanceado donde las alturas de los subárboles de cualquier nodo tienen diferencias en no más de uno

árbol rojo-negro: un árbol binario de búsqueda balanceado con nodos adicionales que indican su color (rojo o negro) para asegurar el equilibrio

**Recorridos de Árboles**

**preorden (nlr): nodo, izquierda, derecha**

def preorden(self, nodo):

if nodo:

print(nodo.valor)

self.preorden(nodo.izquierda)

self.preorden(nodo.derecha)

**inorden (lnr): izquierda, nodo, derecha**

def inorden(self, nodo):

if nodo:

self.inorden(nodo.izquierda)

print(nodo.valor)

self.inorden(nodo.derecha)

**postorden (LRN): izquierda, derecha, nodo**

def postorden(self, nodo):

if nodo:

self.postorden(nodo.izquierda)

self.postorden(nodo.derecha)

print(nodo.valor)