Tema 11: Árboles

Héctor Xavier Limón Riaño

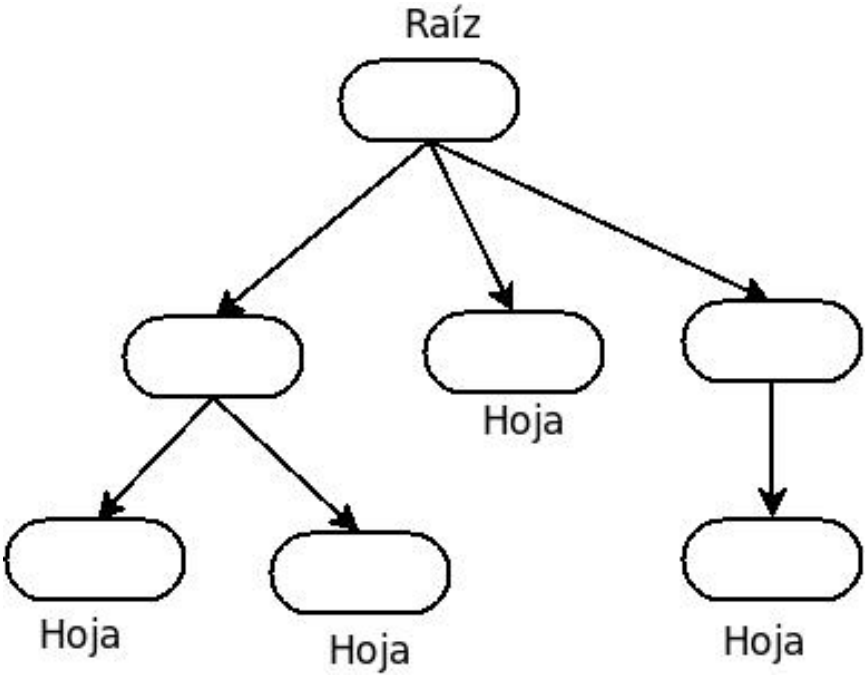
May 31, 2024

Contents

1 Introducción

- En teoría de grafos un arbol es un grafo acıclico dirigido donde cada nodo tiene un solo nodo como padre (exceptuando el nodo raız que no tiene padre)
- Un nodo puede tener cualquier numero de descendientes
- Al primer nodo del arbol se le llama raız (root)
- A traves de la raız se puede acceder a cualquier elemento
- Si un nodo no tiene descendientes se le llama hoja (leaf, leaves)
- Dos nodos con el mismo padre se les llama hermanos (siblings)m
- Un árbol es una estructura recursiva, cada nodo del árbol es a su

vez un árbol



1.1 Aplicaciones

- Algunos ejemplos de aplicaciones:
  - En compiladores (árbol sintáctico)
  - En mineria de datos (arboles de decision)
  - En juegos (alfa-beta)
  - Para busqueda y ordenacion de datos
  - El sistema de archivos se estructura como un árbol
  - DOM HTML/XML: la estructura de formatos de etiquetas sigue una estructura de árbol

2 Tipos

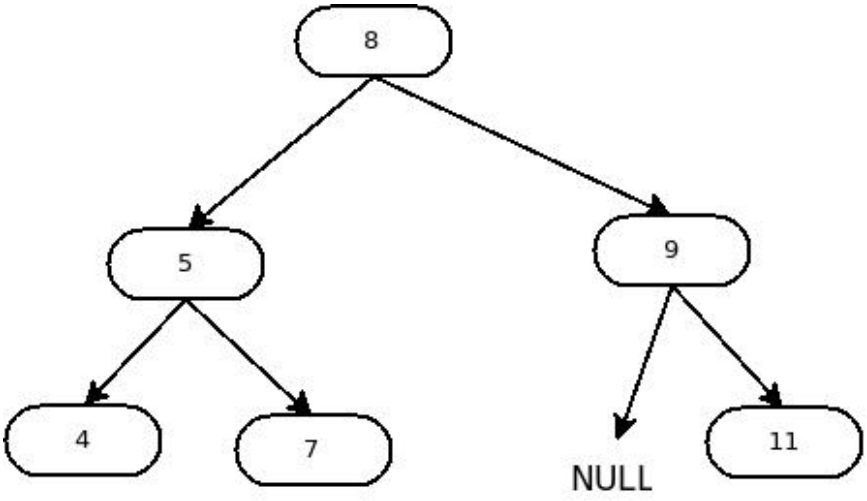
- En este curso se verán dos tipos de árboles:
  - Árboles generales
  - Árboles binarios de búsqueda

2.1 Árboles generales

- Cada nodo puede tener cualquier número de hijos
- No cumplen ninguna propiedad en especial

2.2 Árboles binarios de búsqueda

- Cada nodo puede tener a lo mucho dos hijos
- Están pensados para indexación rápida de datos
- Por ejemplo, se utilizan en bases de datos para recuperar de forma eficiente registros
- Tienen restricciones:
  - Cada nodo tiene un índice numérico
    - El hijo del lado izquierdo tiene un valor de índice menor al padre
    - El hijo del lado derecho tiene un valor de índice mayor al padre



3 Implementación

3.1 Árbol general

- Python no cuenta en su biblioteca estándar con esta estructura de datos
- Se verá una implementación basada en objetos
- La implementación se divide en dos clases:
  - Una clase que represente cualquier nodo del árbol
    - Se tiene una propiedad con los nodos hijos
  - Una clase que representa al árbol en si
    - Se tendrá una propiedad con una referencia al nodo raíz
- Durante el tema se realizará una implementación gradual en clase

3.1.1 Operaciones

- Crear árbol (constructor)
- Agregar hijo
- Es hoja
- Recuperar padre
  - Realiza una modificación a la clase `Nodo` para que esta operación sea sencilla
- Imprimir árbol
  - Se requiere hacer recorridos de árbol
  - Primero en profundidad
    - Recorres siempre primero a la izquierda, hasta llegar a una hoja
    - Al llegar a una hoja regresar un paso, ir a la derecha y continuar por la izquierda
    - El control se puede hacer mediante una pila (con solo recursividad se puede)
  - Primero en amplitud

        * Se recorre el árbol por niveles
        * Hasta terminar de recorrer un nivel se va al siguiente
        * Requiere de una cola para llevar el control
        * Usa mucha más memoria que primero en profundidad

- Borrar hijo (podar)

3.2 Árbol binario de búsqueda

3.2.1 Implementación

- Se implementará mediante dos clases:
  - Nodo
    * 4 propiedades:
      · nodo izquierda
      · nodo derecha
      · índice
      · valor: es lo que se quiere guardar en si, opcional
  - ArbolBinario
    * Sólo se requiere como propiedad la raíz

1. Operaciones de árbol
   - Crear árbol
   - Agregar nodo
   - Recuperar valor
   - Imprimir árbol
   - Borrar nodo