

AED Tema 2. Árboles I: Conceptos básicos

Martín González Dios 

2 de enero de 2025

1. Conceptos básicos

Árbol: **estructura de datos no lineal**, con una organización jerárquica y elementos homogéneos donde cada elemento puede tener varios elementos sucesores, pero un único elemento sucesor. No puede haber rutas circulares en las conexiones de un árbol, de tal forma que sólo puede existir una única ruta desde cada nodo hasta la raíz.

Conceptos:

- **Nodo**: cada uno de los componentes o elementos del árbol.
- **Ascendiente/antecesor**: nodo en un nivel superior dentro de la estructura jerárquica del árbol, accesible por ascenso repetido de hijo a padre.
- **Descendiente/sucesor**: nodo en un nivel inferior dentro de la estructura jerárquica del árbol, accesible por descenso repetido de padre a hijo.
- **Nodo raíz**: nodo inicial del árbol. Se encuentra en el nivel superior de la estructura jerárquica del árbol, nivel 1.
- **Nodo hoja**: nodo terminal del árbol. Son todos aquellos que no tienen descendientes y se encuentran en los niveles más bajos de la estructura jerárquica del árbol, aunque pueden estar en distintos niveles del mismo.
- **Subárbol**: conjunto de nodos de un árbol que a su vez tienen estructura de árbol.
- **Nodo padre**: es el nodo ascendiente directo de sus subárboles. En un árbol cada nodo solo puede tener un padre, excepto el nodo raíz que carece del mismo.
- **Nodo hijo**: nodo descendiente directo de otro nodo. En un árbol cada nodo puede tener varios hijos.
- **Nodos hermanos**: dos nodos son hermanos si tienen el mismo padre.
- **Nodo interno**: es todo nodo que no es ni raíz ni hoja.
- **Camino**: secuencia de nodos donde cualquier par de nodos consecutivos son padre e hijo.
- **Longitud del camino**: número de nodos que forman el camino menos uno.
- **Rama**: cualquier camino desde el nodo raíz hasta un nodo hoja.
- **Nivel (profundidad) del nodo**: número de nodos que tiene el camino desde la raíz a dicho nodo.
- **Grado del nodo**: número de descendientes directos de dicho nodo (número de hijos).
- **Grado de un árbol**: máximo grado de sus nodos.
- **Peso**: número de nodos terminales u hojas del árbol.
- **Altura**: nivel más alto del árbol (número de niveles del árbol). La altura es igual al número de nodos que tienen la rama más larga del árbol.

2. Árboles binarios

Definición: árbol donde cada nodo tiene como máximo grado 2.

Definición recursiva: un árbol binario es un árbol vacío, o bien un nodo raíz con subárboles formados por árboles binarios a la izquierda y a la derecha.

Conceptos:

- **Árbol binario equilibrado:** cuando la diferencia de altura entre los subárboles de cualquier nodo es como máximo una unidad.
- **Árbol binario lleno:** un árbol binario de altura h es lleno si tiene todas sus hojas a nivel h y todos los nodos que están en un nivel menor que h tienen cada uno 2 hijos. (Un árbol binario lleno es totalmente equilibrado)

Un árbol binario lleno es necesariamente completo.

Un árbol binario completo es equilibrado.

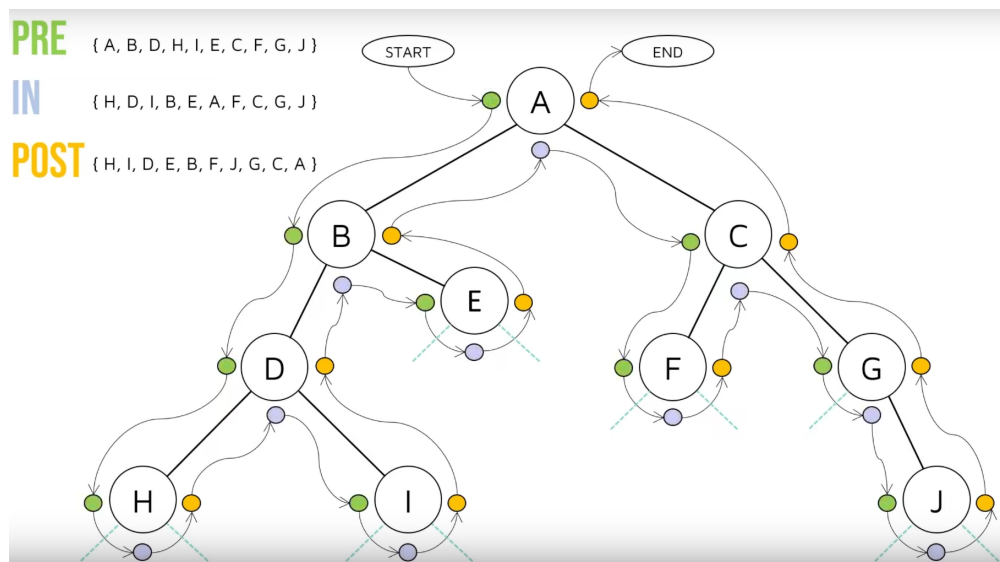
3. Tipos de recorrido de un árbol binario

3.1. Recorrido en anchura

Por niveles y, dentro de cada nivel, de izquierda a derecha.

3.2. Recorrido en profundidad

- **Preorden:** raíz - izq - der. RID, notación prefija o notación polaca.
- **Inorden:** izq - raíz - der. IRD, orden normal de expresiones, orden simétrico u orden central.
- **Postorden:** izq - der - raíz. IDR o notación postfija de expresiones o notación polaca inversa.



3.3. Recorridos no recursivos

Usan estructuras auxiliares para almacenar puntero a los nodos del árbol.

- **Recorrido en profundidad inorden no recursivo.** Usa una **pila** para almacenar los punteros a los nodos del árbol.
 1. Se guardan los elementos izquierdos en la pila hasta llegar a un nodo sin hijo izquierdo.
 2. Se desapila el tope, se imprime y se inserta el hijo derecho.
 3. Se repiten los pasos 1 y 2 hasta que la pila esté vacía.
- **Recorrido en anchura no recursivo.** Usa una cola auxiliar para visitar los nodos por niveles.
 1. Tomar el puntero raíz y ponerlo en la cola.
 2. Quitar el primer elemento de la cola, mostrar el contenido de dicho nodo y almacenar los punteros correspondientes a sus hijos izquierda y derecha.
 3. Repetir el paso 2 hasta que la cola esté vacía y el árbol recorrido.

4. Árboles de expresión

Los árboles binarios se utilizan para **representar expresiones en memoria**, esencialmente en compiladores de lenguajes de programación.

Si suponemos operadores binarios:

- Los paréntesis no se almacenan (implícitos)
- Todos los operandos letras están almacenados en hojas.
- La raíz de cada subárbol es un operador.

Construcción a partir de una expresión en notación convencional (infija):

- **Estructuras de datos auxiliares:**
 - **Pila para almacenar punteros a los nodos de un árbol** (pila del árbol binario).
 - **Pila para retener los operadores** temporalmente hasta que llegue el momento de incorporarlos al árbol. (pila de caracteres)
- **Pasos:**
 1. Cuando se lee un operando se crea un árbol de un nodo y se mete el puntero a él en la pila de punteros.
 2. Cuando se lee un operador:
Si su prioridad \leq que la del tope de la pila de operadores, se desapilan sucesivamente los operadores hasta que ya no se cumpla esta condición.
Cada vez que se saca un operador de la pila también se sacan 2 punteros de la pila de punteros y se usan estos 3 elementos para crear un nuevo árbol. El puntero a este nuevo árbol se coloca en la pila de punteros.
Finalmente, se introduce el operador en la pila.
 3. El proceso termina cuando se acaba la entrada y, en caso de que no esté vacía, se vacíe la pila de operadores.