

Algoritmos y estructuras de datos

Árboles y aplicaciones

Francisco Javier Zaragoza Martínez

Universidad Autónoma Metropolitana Unidad Azcapotzalco
Departamento de Sistemas



28 de agosto de 2024

Miguel de Unamuno

Hubo **árboles** antes de que hubiera libros, y acaso cuando acaben los libros continúen los **árboles**. Y acaso llegue la humanidad a un grado de cultura tal que no necesite ya de libros, pero siempre necesitará de **árboles**, y entonces abonará los **árboles** con libros.

Jacinto Benavente

En cuestión de **árboles** genealógicos es más seguro andarse por las **ramas** que buscar las **raíces**.

Árboles enraizados

Definición recursiva

Un árbol enraizado A puede:

- 1 estar vacío o
- 2 consistir de un nodo r , llamado la raíz de A , y una secuencia de $d_r \geq 0$ árboles enraizados $A_1, \dots, A_{d_r}$ con raíces $r_1, \dots, r_{d_r}$.

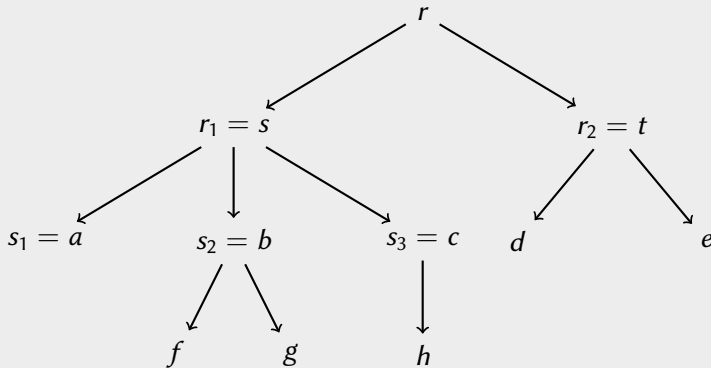
Nodos precursores y sucesores

- 1 El orden de A es la cantidad n de nodos que contiene. Sólo nos interesan los árboles enraizados de orden finito.
- 2 Un nodo s es el precursor de $s_1, \dots, s_{d_s}$ y éstos son sus sucesores.
- 3 Dos nodos que tengan el mismo precursor se dicen semejantes.
- 4 Una hoja de A es un nodo de A sin sucesores.

Árboles enraizados

Ejemplo

Este árbol enraizado A de orden 11 tiene raíz r con $d_r = 2$ y sucesores r_1, r_2 . El nodo s es el precursor de s_1, s_2, s_3 . Los nodos a, d, e, f, g, h son las hojas de A .



Árboles enraizados

Grado, altura y profundidad de un nodo

Grado

El **grado** de un nodo s es la cantidad d_s de sucesores que tiene.

Altura

La **altura** h_s de un nodo s se define de forma recursiva:

- 1 Si s es una hoja, entonces $h_s = 1$.
- 2 En caso contrario $h_s = 1 + \max_{1 \leq i \leq d_s} h_{s_i}$.

Profundidad

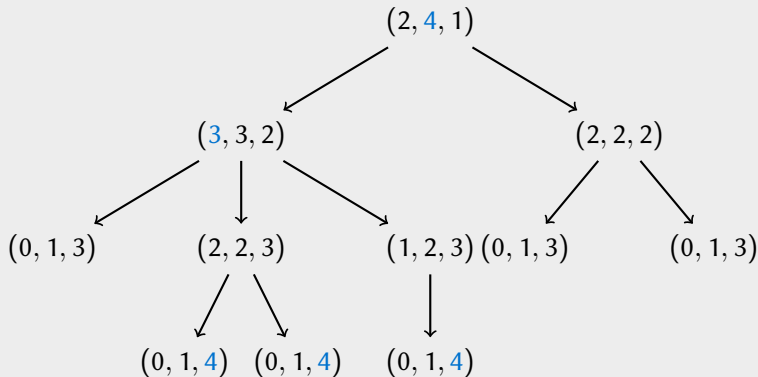
La **profundidad** p_s de un nodo s se define de forma recursiva:

- 1 Si s es la raíz, entonces $p_s = 1$.
- 2 En caso contrario s tiene precursor s' y $p_s = 1 + p_{s'}$.

Árboles enraizados

Un árbol enraizado de grado 3 y altura 4

En cada nodo s anotamos (d_s, h_s, p_s) .



Árboles enraizados

Grado, altura y profundidad de un árbol

En un árbol enraizado

El grado, la altura y la profundidad de A se definen como el máximo grado, la máxima altura y la máxima profundidad de sus nodos, respectivamente.

Altura y profundidad

La altura de un nodo s es la **máxima** cantidad de nodos en un camino de s a alguna hoja. La profundidad de un nodo s es la cantidad de nodos en **el camino** de la raíz a s . La altura y la profundidad de un árbol **coinciden**.

Niveles

Todos los nodos a la misma profundidad conforman un **nivel**. La raíz está a nivel 1, sus sucesores a nivel 2 y así sucesivamente.

Árboles enraizados

Algunas aplicaciones de árboles enraizados

Árboles de recursión

La llamada inicial es la raíz, las llamadas recursivas son sus sucesores, los casos base son las hojas. El tiempo de ejecución es **proporcional** al orden.

Listas enlazadas

Una lista enlazada con $n \geq 1$ nodos es un árbol enraizado de grado 1, altura n y una hoja (el único nodo sin nodo siguiente).

Montículos

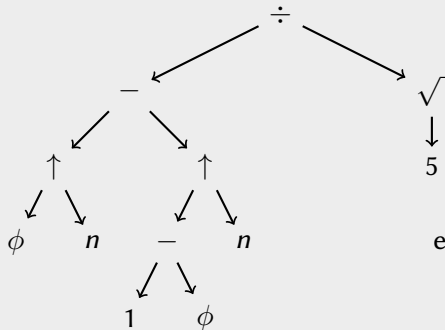
Un montículo con $n \geq 1$ nodos es un árbol enraizado de grado ≤ 2 , altura $1 + \lfloor \log_2 n \rfloor$ y $\lceil \frac{n+1}{2} \rceil$ hojas. Cada precursor tiene un dato mayor o igual al de sus sucesores. Además, los primeros $\lfloor \log_2 n \rfloor$ niveles están llenos.

Árboles enraizados

Ejemplo de árbol de expresión aritmética

Una expresión aritmética consta de una constante o bien de un operador junto con sus operandos, que a su vez son expresiones aritméticas.

Ejemplo (Una fórmula para los números de Fibonacci)



Si $\phi = \frac{1}{2}(1 + \sqrt{5})$,

$$\text{entonces } F_n = \frac{\phi^n - (1 - \phi)^n}{\sqrt{5}}.$$

Árboles enraizados

Evaluación recursiva de expresiones aritméticas

Caso base

El nodo es una hoja del árbol de expresión, por lo que debe ser una constante. El valor de este nodo es el valor de la constante.

Caso recursivo

El nodo tiene sucesores, por lo que debe ser un operador. El valor de este nodo es el operador aplicado al valor de sus sucesores.

Recorrido en posorden

El valor de un nodo sólo se puede obtener después de conocer el valor de sus sucesores. Esta idea (primero hacer algo con los sucesores para luego hacerlo con el precursor) se llama **recorrido en posorden**.

Árboles enraizados

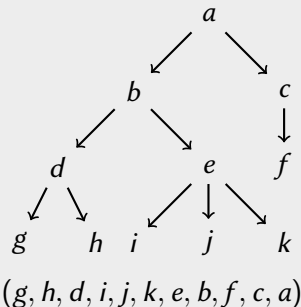
Recorrido en posorden

El algoritmo `posorden(s)` se expresa fácilmente de forma recursiva:

- 1 Para $1 \leq i \leq d_s$
 - 1 `posorden(s[i])`
- 2 `procesa(s)`

Observa que si `s` es una hoja, entonces el primer paso no hace ninguna llamada recursiva, es decir, `s` es un caso base.

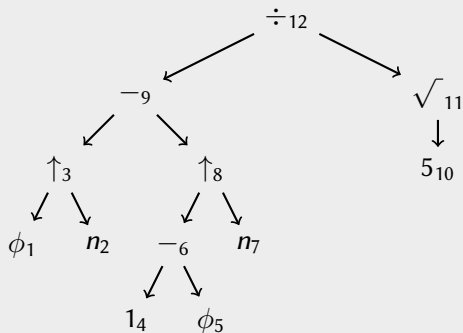
Ejemplo



Árboles enraizados

Ejemplo de evaluación árbol de expresión aritmética

En cada nodo se anotó como subíndice en qué orden se procesó.



Al orden $\phi_1, n_2, \uparrow_3, 1_4, \phi_5, -6, n_7, \uparrow_8, -9, 5_{10}, \sqrt{11}, \div_{12}$ también se le llama notación **polaca inversa**, **posfija** o **sufija**.

Árboles enraizados

Árboles de directorios

Sistemas de archivos

Un sistema de archivos tiene un directorio raíz que consta de una secuencia de archivos, algunos de los cuales son a su vez directorios.

Copia de sistemas de archivos

Primero se crea la copia del directorio raíz y luego se copia recursivamente la secuencia de archivos.

Recorrido en preorden

La secuencia de archivos sólo se puede copiar después de haber creado el directorio donde están. Esta idea (primero hacer algo con el precursor para luego hacerlo con sus sucesores) se llama **recorrido en preorden**.

Árboles enraizados

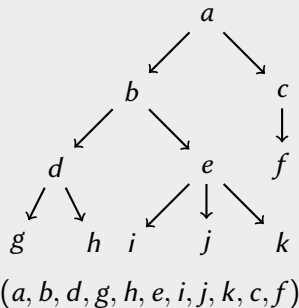
Recorrido en preorden

El algoritmo `preorden(s)` se expresa fácilmente de forma recursiva:

- 1 `procesa(s)`
- 2 Para $1 \leq i \leq d_s$
 - 1 `preorden(s[i])`

Observa que si `s` es una hoja, entonces el segundo paso no hace ninguna llamada recursiva, es decir, `s` es un caso base.

Ejemplo



Árboles enraizados

Recorrido con pilas

Considera una **pila** de nodos que inicia vacía y este algoritmo:

- 1 Apila la raíz r de un árbol enraizado A .
- 2 Mientras la pila no esté vacía: Desapila s y apila $s_{d_s}, \dots, s_1$.

No es difícil convencerse de que el orden de desapilado es el **preorden** de A . Esto nos permite una implementación **no recursiva** de preorden.

Árboles enraizados

Recorrido con colas

Considera ahora una **cola** de nodos que inicia vacía y este algoritmo:

- 1 Forma la raíz r de un árbol enraizado A .
- 2 Mientras la cola no esté vacía: Desforma s y forma $s_1, \dots, s_{d_s}$.

No es difícil convencerse de que el orden de desformado nos entrega un **recorrido por niveles** de A comenzando por su raíz.