Algoritmos y Estructuras de Datos II

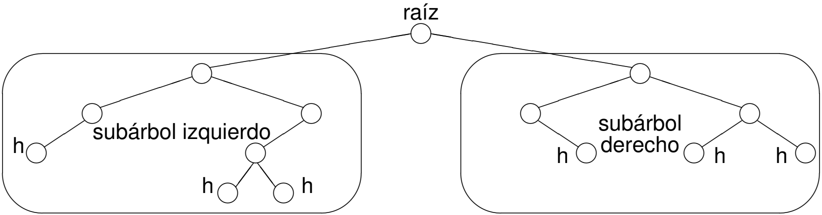
Árboles binarios de búsqueda

6 de mayo de 2019

Clase de hoy

1. Árboles binarios
   - Especificación
   - Terminología habitual
   - Posiciones
2. Árbol binario de búsqueda
   - Ejemplos y definiciones

Intuición



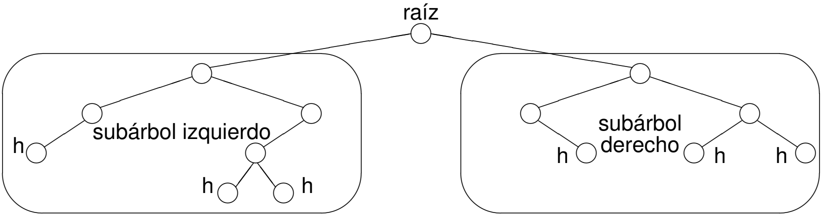
Todos los árboles pueden construirse con los constructores

- $<>$, que construye un árbol vacío
- $<\_,\_,\_>$, que construye un árbol no vacío a partir de un elemento y dos subárboles

Notación $<>$

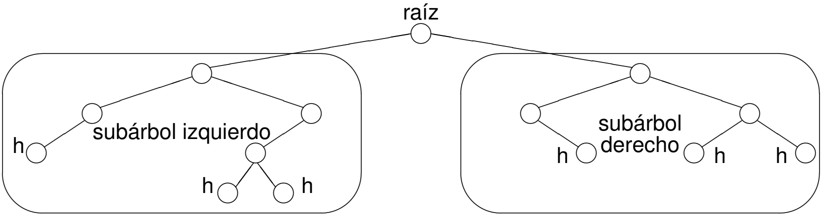
- Notar la sobrecarga de la notación $<>$:
  - $<>$ es el árbol vacío,
  - $<i,r,d>$ es el árbol no vacío cuya raíz es r, subárbol izquierdo es i y subárbol derecho es d.
  - $<r>$ es la hoja $<<>,r,<>>$
- Conclusión: la notación $<>$ puede tener 0, 1 ó 3 argumentos.

Botánica y genealogía



- Un **nodo** es un árbol no vacío.
- Tiene **raíz**, **subárbol izquierdo** y **subárbol derecho**.
- A los subárboles se los llama también **hijos** (izquierdo y derecho).
- Y al nodo se le dice **padre** de sus hijos.
- Una **hoja** es un nodo con los dos hijos vacíos.

Más terminología



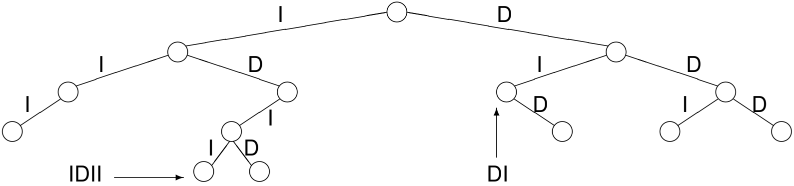
Terminología:

- Se usa terminología genealógica como **hijo, padre, nieto, abuelo, hermanos, ancestro, descendiente**.
- También de la botánica: **raíz, hoja**.
- Se define **camino, altura, profundidad, nivel**.

Sobre los niveles

- En el nivel 1 hay a lo sumo 1 nodo.
- En el nivel 2 hay a lo sumo 2 nodos.
- En el nivel 3 hay a lo sumo 4 nodos.
- En el nivel 4 hay a lo sumo 8 nodos.
- En el nivel i hay a lo sumo 2^{i-1} nodos.
- En un árbol de altura n hay a lo sumo $2^0 + 2^1 + \ldots + 2^{n-1} = 2^n - 1$ nodos.
- En un árbol “balanceado” la altura es del orden del $\log_2 k$ donde k es el número de nodos.

Indicaciones/posiciones



$<> \downarrow p = <>$
$<i,e,d> \downarrow \mathrm{R} = <i,e,d>$
$<i,e,d> \downarrow (\mathrm{I}\ \mathrm{p}) = i \downarrow p$
$<i,e,d> \downarrow (\mathrm{D}\ \mathrm{p}) = d \downarrow p$

$<i,e,d>.\mathrm{R} = \mathrm{e}$
$<i,e,d>.(\mathrm{I}\ \mathrm{p}) = \mathrm{i.p}$
$<i,e,d>.(\mathrm{D}\ \mathrm{p}) = \mathrm{d.p}$

o equivalentemente $t.p = raiz\ (t \downarrow p)$.
Se define $pos(t) = \{p \in pos \mid t \downarrow p \neq <>\}$. Es el conjunto de las posiciones del árbol binario t.

Árboles binarios de búsqueda

- Son casos particulares de árboles binarios,
- son árboles binarios *t* en donde la información está organizada de forma tal que el siguiente algoritmo sencillo permite buscar eficientemente un elemento:
- el elemento buscado se compara con la raíz de *t*
  - si es el mismo, la búsqueda finaliza
  - si es menor que la raíz, la búsqueda se restringe al subárbol izquierdo de *t* con el mismo algoritmo
  - si es mayor que la raíz, la búsqueda se restringe al subárbol derecho de *t* con el mismo algoritmo.
- Si el árbol está “balanceado”, es un algoritmo logarítmico.

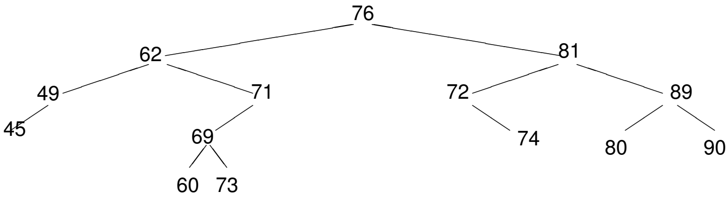
Definición intuitiva

Para que este algoritmo funcione, *t* debe cumplir lo siguiente:

- los valores alojados en el subárbol izquierdo de *t* deben ser menores que el alojado en la raíz de *t*,
- los valores alojados en el subárbol derecho de *t* deben ser mayores que el alojado en la raíz de *t*,
- estas dos condiciones deben darse para todos los subárboles de *t*.

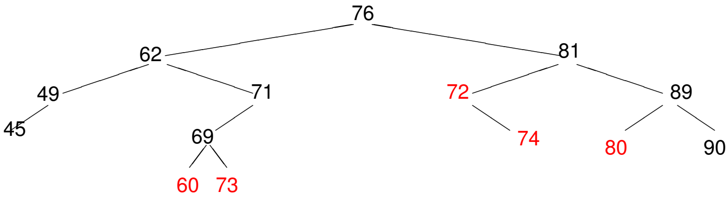
Si se cumplen estas condiciones, decimos que *t* es un **árbol binario de búsqueda** o **ABB**.

Ejemplo



¿Es un árbol binario de búsqueda?

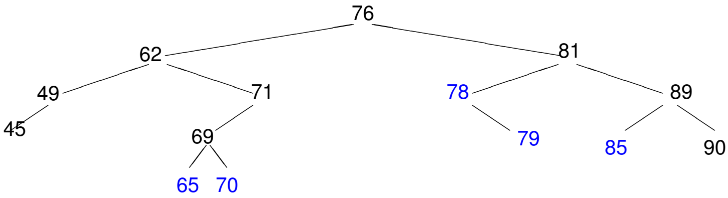
Ejemplo



No es un árbol binario de búsqueda.

- 60 debe cambiar por uno entre 63 y 68
- 72 debe cambiar por uno entre 77 y 80
- 73 debe cambiar por 70
- 74 debe cambiar por uno entre 77 y 80.
- 80 debe cambiar por uno entre 82 y 88.

Ejemplo



Ahora sí es un árbol binario de búsqueda.