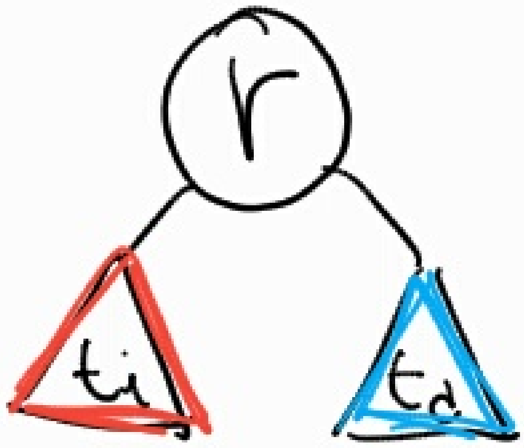
ABB



$t_i < r$
$t_d > r$

es una def recursiva (ti y td son arboles que deben cumplir lo mismo)

- Buscar
- Insertar
- Borrar

Temp: $O(n)$

se ve que el peor caso es lo mismo que operar en una lista/arreglo/etc.

aparecen los BALANCEADOS

→ AVL

AVL

Balance por altura

→ son ABB

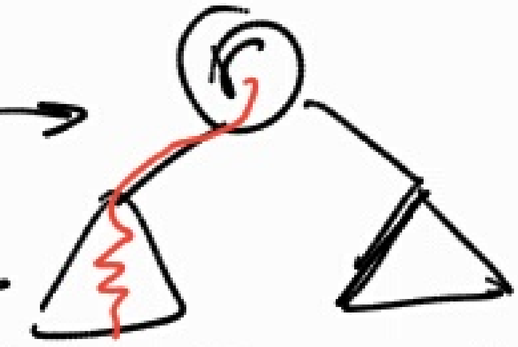
→ $|h(t_i) - h(t_d)| \leq 1$ → diferencia es 0, 1 o -1

→ t_i y t_d son AVL

Prop: Si t es un AVL de n nodos $\Rightarrow h(t) \in \Theta(\log n)$

- Sea t un AVL de altura $h \Rightarrow |t| \geq |F_h|$

• Buscar → Tiempo $O(\log(n))$ →



el peor caso sería recorrer todos los niveles sin encontrar el elemento

* Insertar
* Borrar

→ $O(\log(n))$

en general se deben aplicar Rotaciones

!

• Factor de balance (FB) → 3 posibilidades

↳ 0 → $h(t_i) = h(t_d)$

↳ -1 → $h(t_i) = h(t_d)+1$

↳ 1 → $h(t_i) = h(t_d)-1$

Altura_AB(t) → Temp: O(n)

CODIGO

Alt(t): si t es vacío (DEVOLVER: 0)

DEVOLVER: MAX(Alt(ti), Alt(td)) + 1.

Altura_AVL(t) → temp O(log(n))

Alt(t):

si t es vacío (DEVOLVER: 0)

si FB = 1 (DEVOLVER 1 + Alt(ti))

si FB = -1 (DEVOLVER 1 + Alt(td))

si FB = 0 (DEVOLVER 1 + Alt(td))

Guía 7
3

14) Tengo un arreglo con un PREORDER AVL y un valor e. Hacer una función que diga si e se encuentra en el árbol (Restricción temp < O(n))

- PREORDER → | r | ti | td |

$<r$ $>r$

de esa condición puedo pensar en usar la BÚSQUEDA BINARIA (MODIFICADA)

si e=r ✓
si e<r
si e>r
→ Necesitamos conocer el límite entre ti ∧ td

OBS: si tuviera el ARBOL AVL la operación sería TRIVIAL O(log(n))

Pero NO vale la pena reconstruirlo, ya que sería mínimo O(n)

↓ PSEUDOCÓDIGO:

- Complejidad temp →

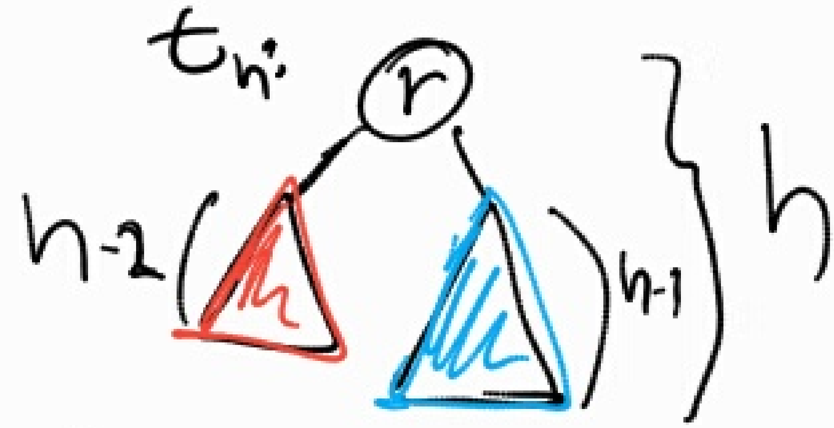
$$T(n) \leq \underbrace{\Theta(\log(n))}^{\text{BUSQUEDA BINARIA}} \cdot \underbrace{h(t)}_{\Theta(\log(n))}$$

$$\therefore \boxed{\Theta(\log^2(n))} < \Theta(n) \checkmark$$

Ejercicio de Parcial → VERDADERO Y FALSO

i) La cantidad de hojas de cualquier ABB de n nodos es $\Theta(n)$. **F** → contraejemplo

ii) Dado un AVL, la diferencia de tamaño (nodos) entre sus árboles es cte. **F**



$$\big|\,|A_{h-1}| - |A_{h-2}|\,\big| = \big|[2^{h-1}-1] - [2^{h-2}-1]\big| = 2^{h-2}(2-1)$$

$A_h = 2^h - 1$

$$\boxed{= 2^{h-2}}$$

↳ crece con h

$\Rightarrow \neq O(1) \therefore F$

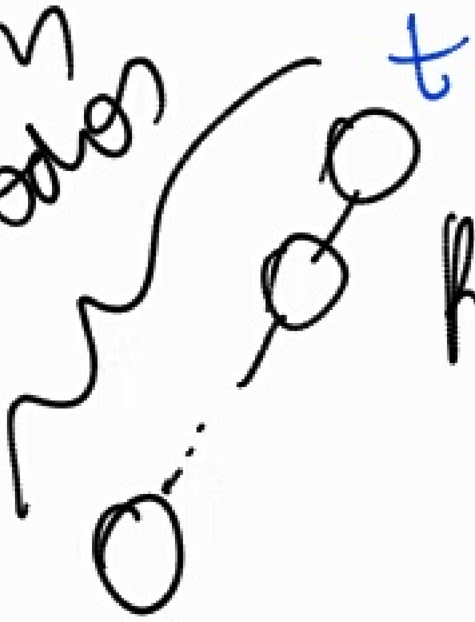
iii) Sea t un AB de n nodos y $k_i(t)$ la # nodos en el i-esimo nivel de t

$\Downarrow$

es posible encontrar un i / $k_i(t) \in \Theta(n)$

$\downarrow$ F

n nodos



$k_i(t) = 1$

$\forall i \in \Omega(n)$