

Laboratorio Nro. 4 Tablas de Hash y Árboles

Stiven Yepes
Universidad Eafit
Medellín, Colombia
syepes@eafit.edu.co

Sara Rodríguez
Universidad Eafit
Medellín, Colombia
srodriguev@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Expliquen qué tipo de árbol utilizaron para representar el sistema de archivos y qué complejidad tiene la operación de búsqueda en él.

El árbol utilizado para representar el sistema de archivos es uno llamado "JTree", el cual también se puede ver como una jerarquía. En este árbol la operación de búsqueda tiene una complejidad de $O(n)$ en el peor de los casos, donde n es el número de archivos o directorios.

3.2 ¿Se puede implementar más eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico? ¿O no se puede? ¿Por qué?

>Si el árbol está balanceado se pueden lograr estos tiempos, por lo cual es necesario asegurarse que las alturas de los árboles sean transformadas para mantener una estructura pareja.

En un árbol bien balanceado:

Operación	Big-O
Lookup - Búsqueda	$O(\log n)$
Insertion - Inserción	$O(\log n)$
Removal - Borrado	$O(\log n)$
In-order traversal – Recorrido In-order	$O(n)$

3.3

Se crea un stack para guardar los datos a medida que se procesan, se empuja la raíz al stack y se itera en los demás nodos del arreglo, y evalúa si el elemento siguiente al actual es menor o mayor, así lo clasifica acorde a su valor numérico.

3.4

ConstruirArbol -> A primera vista parece un $O(n^3)$ teórico al ver los métodos generalmente, sin embargo, teniendo en cuenta los valores y como se construye el árbol las operaciones de push/pop usadas tienen $2n$ de complejidad máxima. Por lo cual la complejidad real del ejercicio es de $O(n)$.

PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

Código ST0245

ImprimirInorder -> $O(n)$

3.5

La n se refiere a la longitud del arreglo que representa la longitud del árbol impreso en pre-order.

4) Simulacro de Parcial

4.1 (b) que inician con la misma letra colisionan
(d) $O(1)$

4.2 c) 3

4.3 Línea 3: return false;

Línea 5: return suma == a.dato;

Línea 7: return sumaElCamino(a.izq, suma - a.dato)

Línea 8: || sumaElCamino(a.der, suma - a.dato);

4.5 a) toInsert == null

b) toInsert <= p

4.7 a) 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
b) 2

4.8 c) 4

4.9 a) 5, 3, 6, 1, 7, 4, 8, 0, 2

4.10 a) Sí

4.11 b) 2, 3, 4, 0, 5, 7, 6
a) 5
B) No

4.12 i) A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8, I = 9, J = 10.
a) G, D, B, A, C, E, F, I, H, J
a) $O(n)$

4.13 raiz.id

a) $T(n) = T(n - 1) + c$, que es $O(n)$

5) Lectura recomendada (opcional)

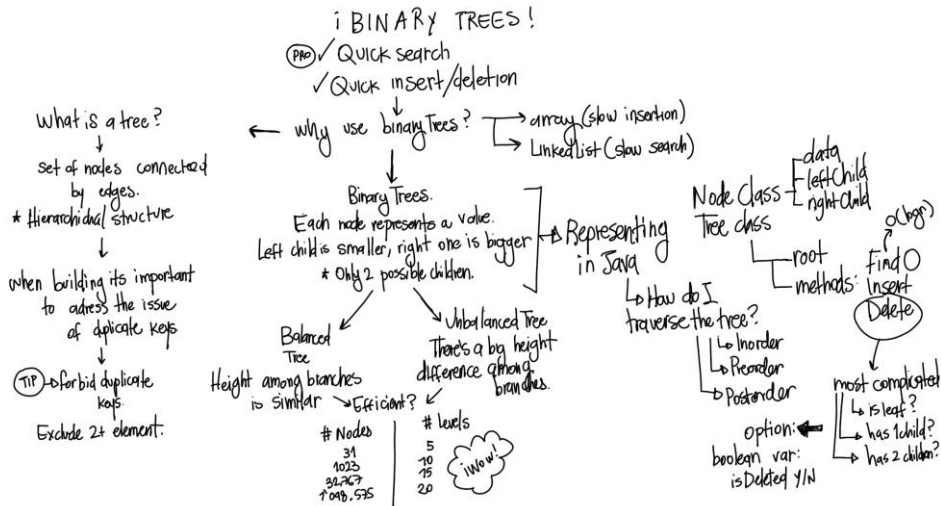
PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1

Código ST0245



Full-size:

https://eafit-my.sharepoint.com/:i:/g/personal/srodriguev_eafit_edu_co/EfBW8tbpzHBBgD6QNZtT64cB-6-CwMyYQkNHVEp4ESjpoA?e=vJ1cDi

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473