

Laboratorio Nro. 4 Tablas de Hash y Árboles

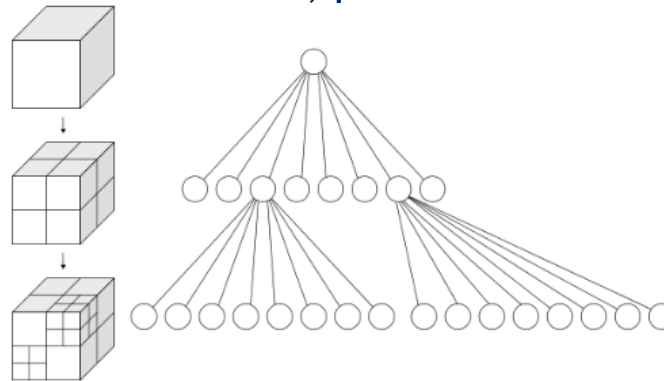
Isabella Echeverri Villa
Universidad Eafit
Medellín, Colombia
iecheverrv@eafit.edu.co

Samuel Arturo Flórez Rincón
Universidad Eafit
Medellín, Colombia
saflorezr@eafit.edu.co

Laura Echavarría Peláez
Universidad Eafit
Medellín, Colombia
lechavarrp@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Utilizamos un árbol octal o un octree, que se ve así:



Como se observa en la imagen, es una estructura muy conveniente para realizar este algoritmo puesto que va dividiendo el espacio en 8 (por esto se llama árbol octal) según las características de cada subdivisión, por ejemplo, si en el espacio total hay cierta cantidad de abejas, va a dividir ese espacio de manera que, al final, cada abeja quede en un subespacio. La complejidad del algoritmo que analiza la posibilidad de que una abeja colisione con otra es de $O(n)$.

3.2 Podríamos considerar la posibilidad de en vez de utilizar un árbol binario de búsqueda, utilizar un árbol binario de búsqueda auto balanceado (con el fin de mejorar la complejidad en tiempo del algoritmo), sin embargo, para el caso específico de un árbol genealógico, no tiene mucho sentido organizar el árbol de esta manera, pues quedaría deformada la posición de los miembros del árbol.

3.3 Para realizar el ejercicio 2.1 hay dos métodos necesarios, el primero es el que recibe el arreglo con los datos (organizados en preorder) que necesitan ser puestos en el árbol de búsqueda. Para realizar este método solo es necesario poner la raíz (que es el primer elemento del arreglo) como la raíz del árbol y posteriormente, hacer un ciclo que llame al

ESTRUCTURA DE DATOS 1

Código ST0245

método agregar con cada numero siguiente del arreglo. El segundo método es el que va a imprimir el árbol en postorder, en este orden se imprime primero la parte de la izquierda, luego la derecha y, por último, la raíz.

3.4 La complejidad del método que imprime en postorder tiene una complejidad de:

$T(n) = T(n/2) + T(n/2) + c$ que es $O(n)$

La complejidad del método que pone los números en el árbol a partir del preoder tiene una complejidad de:

$O(n)$

3.5 La variable n es el tamaño del árbol.

4) Simulacro de Parcial

4.1 B. Las cadenas que inician con la misma letra colisionan

La complejidad asintótica es $O(1)$

4.2 1. Retorna el ancestro común más lejano

2. $T(n) = T(n/2) + T(n/2)$ que es $O(n)$

3. Haciéndolo un árbol binario de búsqueda auto balanceado

4.3 Return True

$O(n+m)$

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

4.4.2 a) $O(n)$

4.4.3 a) Wilkenson, Sufranio, Piolín, Usnavy, Piolina, Wilberta, Joaquina, Eustaquio, Florinda, Eustaquia, Yovín

4.4.4 b) Cambiar el orden de las líneas 03, 04 y 05 por 04, 05, 03

4.5 Línea 04 toinsert == p

Línea 06 toinsert > p

5) Lectura recomendada (opcional)

Mapa conceptual

PhD. Mauricio Toro Bermúdez

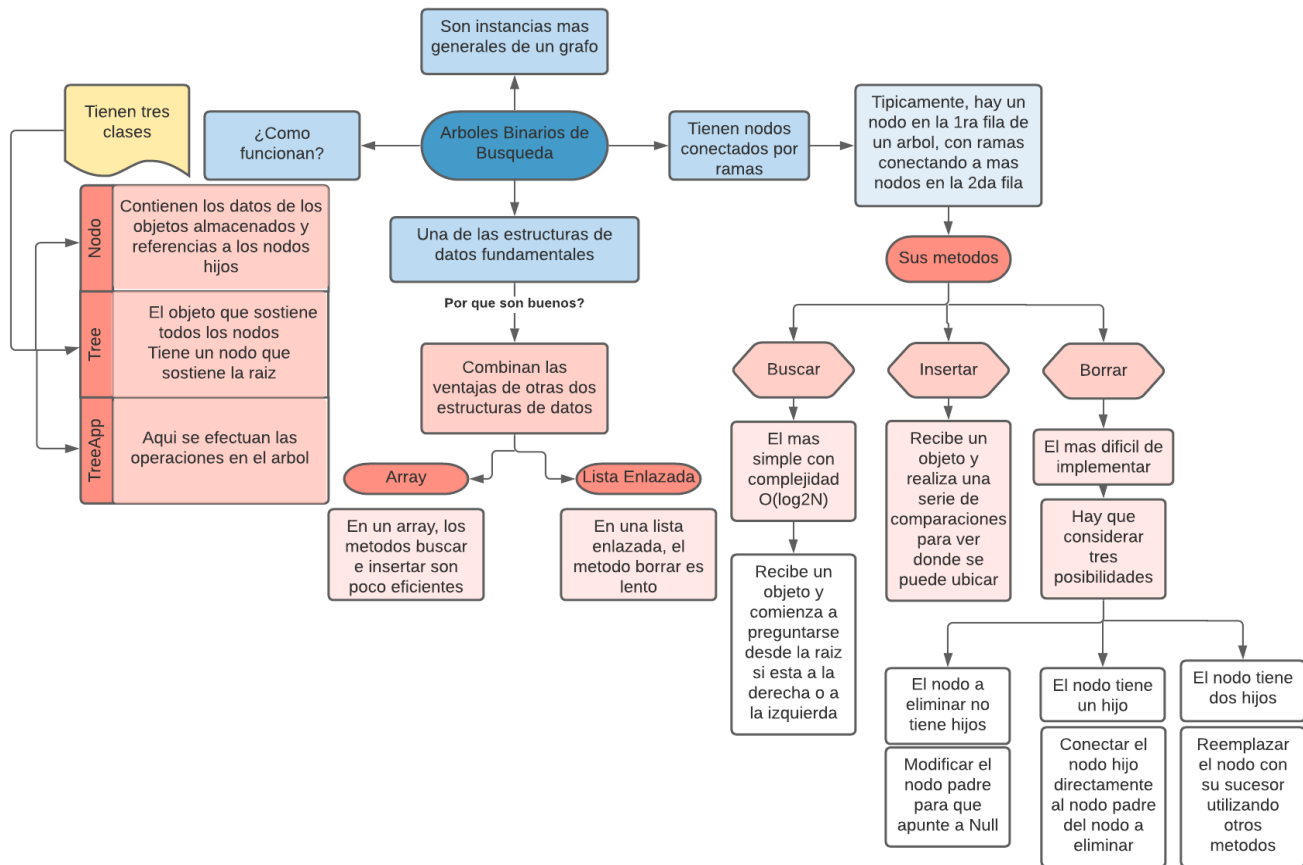
Docente | Escuela de Ingeniería | Informática y Sistemas

Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627

Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

Código ST0245



6) Trabajo en Equipo y Progreso Gradual (Opcional)

6.1 Actas de reunión

6.2 El reporte de cambios en el código

6.3 El reporte de cambios del informe de laboratorio

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