

## Laboratorio Nro. 4

### Tablas de Hash y Arboles

Federico Vélez Quintero  
Universidad Eafit  
Medellín, Colombia  
fvelezq@eafit.edu.co

Manuela Franco  
Universidad Eafit  
Medellín, Colombia  
mfrancoo1@eafit.edu.co

José David Gómez  
Universidad Eafit  
Medellín, Colombia  
Jdgomezz@eafit.edu.co

Rafael Villegas  
Universidad Eafit  
Medellín, Colombia  
Correointegrante2@eafit.edu.co

### 3) Simulacro de preguntas de sustentación de Proyectos

**3.1** Para esta estructura quisimos utilizar un árbol octal, ya que es una opción eficiente para calcular colisiones en un espacio tridimensional, es muy útil en este caso porque es una estructura que va dividiendo el espacio en ocho según la cantidad de objetos que le indiquemos, en este caso, por ejemplo, todas las abejas quedaran en un subespacio específico.

La estructura es muy eficiente, especialmente en este caso, entonces la complejidad de que nos dice la posibilidad de que las abejas colisionen es de  $O(n)$ .

**3.2** No se puede implementar eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico, debido a que, quedaría distorsionada la posición de los miembros del árbol afectando el orden genealógico, lo cual no tendría sentido y, además, para que el árbol mejore la complejidad y se aminore sería necesario que el árbol fuera auto balanceado.

**3.4**  $T(n) = T(n/2) + T(n/2) + C$   
 $= O(n)$

**3.5**  $n$  es la variable de la cantidad de nodos que construye un árbol.

### 4) Simulacro de Parcial

**4.1**

4.1.1 b

4.1.2 d

**4.2**

1. Devuelve el menor antecesor común al nodo 1 y al nodo 2

**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

2.  $O(n)$

3. Podemos mejorar el algoritmo *mystery* utilizando la condición de que si  $n$  es menor que los datos se asignarán al nodo de la izquierda; si no, al nodo de la derecha, así evitaremos pasar por los dos lados.

**4.3** 1

2  $O(n)$

**4.4**

4.4.1 C

4.4.2 A

4.4.3 D

4.4.4 A

**4.5**

A)  $toInsert == p.data$

B)  $toInsert > p.data$

**4.6**

4.6.1 D

4.6.2 Return 0

4.6.3

**4.7**

4.7.1 A

4.7.2 B

**4.8** B

**4.9** A

**4.10**

4.10.1 B

4.10.2 A

4.10.3 B

**4.11** 4.11.1 *raiz.id*

4.11.2 A

**4.12** 4.12.1 *i*

4.12.2 *i*

4.12.3 *a*

**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