

Laboratorio Nro. 4 Tablas de Hash y Árboles



Objetivos: 1. Escoger la estructura de datos apropiada para modelar un problema dado. 2. Solucionar problemas del mundo real con algoritmos.



Consideraciones: Lean y verifiquen las consideraciones de entrega,



Leer la Guía



Trabajo en
Parejas



Si tienen reclamos,
regístrenlos en
<http://bit.ly/2g4TTKf>



Ver
calificaciones
en Eafit
Interactiva



En el GitHub
docente,
encontrarán la
traducción de
los Ejercicios
en Línea

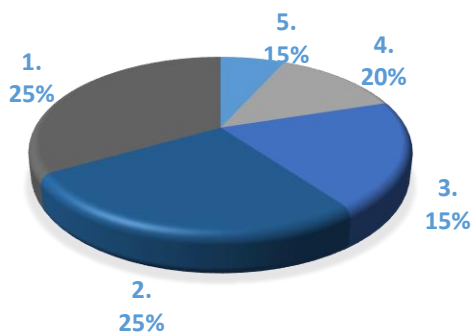


Hoy, plazo de
entrega



Subir el **informe pdf** en la carpeta **informe**, el **código del ejercicio 1** en la carpeta **codigo** y el **código del 2** en la carpeta **ejercicioEnLinea**

Porcentajes y criterios de evaluación



- 1. Simulacro sustentación proyecto
- 2. Análisis de complejidad
- 3. Código de laboratorio
- 4. Simulacro de parcial
- 5. Ejercicio con juez en línea

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

Resumen de Ejercicios a Resolver

1.1 Implementen un algoritmo para identificar las abejas robóticas que se encuentren a 100 metros o menos de distancia de otra abeja.

1.2 [Ejercicio Opcional] Desarrollar una estructura de datos para representar los directorios y archivos en un sistema de archivos, y consultar, eficientemente, los archivos y subdirectorios que se encuentren en un directorio.

1.3 [Ejercicio opcional] Escriban un método para la clase `BinaryTree` que calcule quién es la abuela materna de una persona.

2.1 Resuelvan el problema usando árboles binarios

2.2 [Ejercicio Opcional] Resuelvan el siguiente problema <http://bit.ly/2iNuHaa>

3.1 Expliquen qué estructura de datos utilizaron para calcular las colisiones entre abejas, por qué la eligieron y qué complejidad tiene el algoritmo que, utilizando dicha estructura de datos, calcula las colisiones.

3.2 [Ejercicio Opcional] 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?

3.3 [Ejercicio Opcional] Expliquen con sus propias palabras cómo funciona la implementación del ejercicio 2.1 y el 2.2

3.4 Calculen la complejidad de los ejercicios 2.1 y 2.2

3.5 Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.3

4. Simulacro parcial

5. [Ejercicio Opcional] Lectura recomendada

6. [Ejercicio Opcional] Trabajo en Equipo y Progreso Gradual

7. [Ejercicio Opcional] Laboratorio en inglés

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

1. Simulacro de Proyecto

Códigos para entregar en GitHub en la carpeta `codigo`

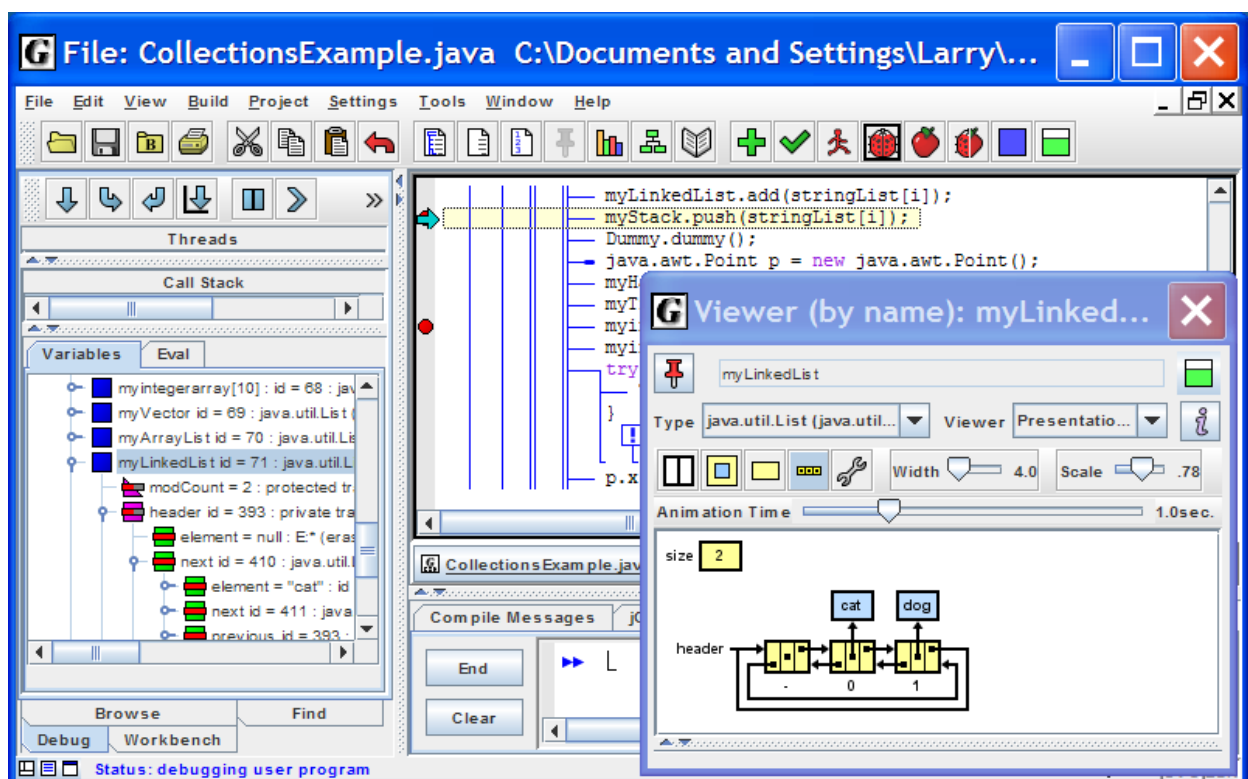
ESTRUCTURA DE DATOS 1

Código ST0245

Resolver ejercicios

Nota 1: Para este taller, de manera especial, se recomienda utilizar el *IDE Jgrasp* (<http://www.jgrasp.org/>) porque tiene un depurador gráfico excelente para estructuras de datos

Nota 2: Vean a continuación gráfica de *Jgrasp* para efectos de ejemplificación



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

1 Simulacro de Proyecto: Códigos para entregar en GitHub en la carpeta `codigo`



En la vida real, la documentación de software hace parte de muchos estándares de calidad como CMMI e ISO/IEC 9126



Vean Guía numeral 3.4



Código del ejercicio en línea en **GitHub**. Vean Guía en numeral 4.24



Documentación opcional. Si lo hacen, utilicen **Javadoc** o equivalente. No suban el HTML a GitHub.



No se reciben archivos en **.RAR** ni en **.ZIP**

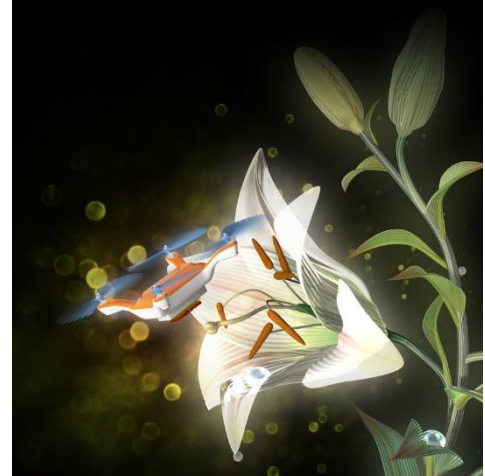


Utilicen Java, C++ o Python

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

1.1 “Cerca de tres cuartos de las especies que se utilizan en cultivos, desde manzanas hasta almendras, necesitan de la polinización de abejas y otros insectos. Infortunadamente, los pesticidas, la deforestación y el cambio climático han causado que disminuya la población de abejas, causando graves problemas a los agricultores. Un dron que pueda polinizar flores puede funcionar, en un futuro no muy lejano, para mejorar el rendimiento de los cultivos.” Un problema que tendremos con la aparición de las abejas robóticas es poder prevenir las colisiones entre ellas.
(<https://www.newscientist.com/article/2120832-robotic-bee-could-help-pollinate-crops-as-real-bees-decline/>).



► Implementen un algoritmo para **identificar las abejas robóticas que se encuentren a 100 metros o menos de distancia de otra abeja.**

La prioridad del algoritmo es la **eficiencia en tiempo de ejecución** y NO el consumo de memoria.

Utilicen los conjuntos de datos que se encuentran en la carpeta **datasets/abejas**, en Github, para probar su algoritmo.



Sea $a = (x, y, z)$ la ubicación geodésica de una abeja robótica, donde x es la latitud en grados, y es la longitud en grados y z es la altura sobre el nivel del mar en metros, y A un conjunto de ubicaciones de las abejas robóticas. Calcule el conjunto de las abejas A' que se encuentran a 100 o menos metros de otra abeja: $A' = \{a | a \in A, \exists b \in A \wedge d(a, b) \leq 100\text{mts}\}$, donde $d(a, b)$ es una función que calcula la distancia en metros entre la ubicación de dos abejas $a, b \in A$.

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

1.2 [Opc] Los avances en la tecnología han hecho que hoy en día sea necesario almacenar y procesar grandes volúmenes de información. Para poder manejar la información en el disco duro se utiliza un sistema de archivos.

Un sistema de archivos es un componente de un sistema operativo encargado de asignar espacio a los archivos, administrar el espacio libre y permitir el acceso a los datos guardados.

Hoy en día existe la necesidad de almacenar y acceder a un número muy grande de archivos en un sistema de archivos. Una de las funciones del sistema de archivos es listar los contenidos de un directorio eficientemente



Extraída de www.google.com

El problema de listar el contenido de un directorio consiste **en encontrar eficientemente, los archivos y subdirectorios que se encuentran en un directorio.**

▶ El objetivo es desarrollar una estructura de datos para representar los directorios y archivos en un sistema de archivos, y consultar eficientemente los archivos y subdirectorios que se encuentren en un directorio.



Ejemplo: Consideren la siguiente estructura de directorios en un sistema de archivos con el formato [usuario tamaño] archivo o directorio:

Proyecto/

```

|— [root 4.0K] DataSets
|   |— [mauriciotoro 252K] treeEtc.txt
|— [mauriciotoro 4.0K] Plantillas
|   |— [mauriciotoro 463K] ED1-Guia-Proyecto-Entrega.doc
|   |— [mauriciotoro 1.5M] ED1-Plantilla-Eafit.pptx
|   |— [mauriciotoro 255K] ejemplo-de-un-reporte-con-plantilla-ACM.pdf
|   |— [mauriciotoro 1.1M] plantilla-ACM-en-Latex.zip
|   |— [root 1.2M] plantilla-ACM-en-Word.doc

```

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

```
| └─ [mauriciotoro 4.0K] SIG Proceedings Template-Jan2015 Zip
|   └─ [mauriciotoro 9.4K] acmcopyright.sty
|   └─ [mauriciotoro 144K] acm-update.pdf
|   └─ [mauriciotoro 336K] flies.eps
|   └─ [mauriciotoro 151K] fly.eps
|   └─ [mauriciotoro 3.4K] rosette.eps
|   └─ [mauriciotoro 65K] sig-alternate-05-2015.cls
|   └─ [mauriciotoro 124K] sig-alternate-guide.docx
|
|   └─ [mauriciotoro 572K] sig-alternate-guide.pdf
|   └─ [mauriciotoro 255K] sig-alternate-sample.pdf
|   └─ [mauriciotoro 26K] sig-alternate-sample.tex
|   └─ [mauriciotoro 1.5K] sigproc.bib
└─ [mauriciotoro 571K] Proyecto Final ED1 Sistema-directorios Vr 2.0.pdf
```

Si el usuario consulta los subdirectorios y archivos que se encuentran en “*Proyecto/DataSets*” la respuesta debe ser “*treeEtc.txt*”.



Como otro ejemplo, si el usuario consulta los archivos y subdirectorios que se encuentran en “*Proyecto/Plantillas/*” y ocupan más de 1 Megabyte, la respuesta debe ser “*ED1-Plantilla-Eafit.pptx*” y “*plantilla-ACM-en-Latex.zip*”.



Como otro ejemplo si el usuario consulta los archivos y subdirectorios que se encuentran en “*Proyecto/*” cuyo dueño sea el usuario root, debe retornar “*DataSets/*” y “*plantilla-ACM-en-Word.doc*”

Finalmente, si el usuario consulta por los subdirectorios y archivos que se encuentran en “*Proyecto/Datos3*” la respuesta debe ser “*No such file or directory*”.

Conjunto de Datos

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

La estructura de directorios se entrega en un archivo formato TXT con el formato que se muestra en el ejemplo, que es el formato que entrega el programa de la línea de comandos `tree -shaul` que está disponible en Windows, Linux y Mac OS.

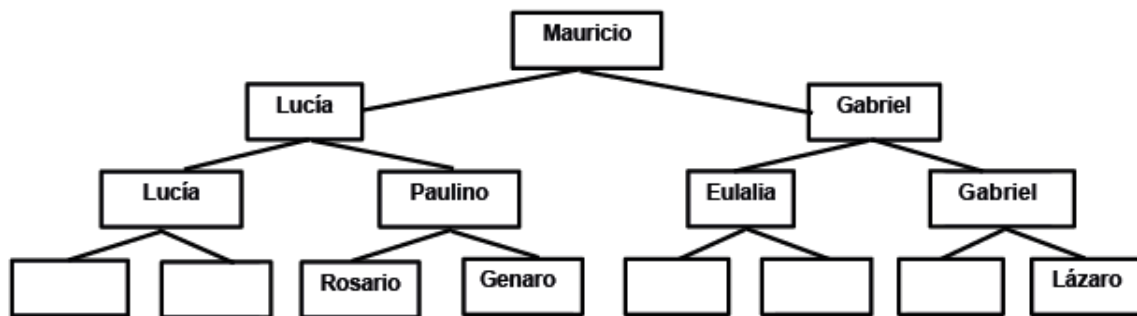
En el *github* hay 3 conjuntos de datos: *ejemplito.txt*, *juegos.txt* y *treeEtc.txt*

1.3 [Opc] Investiguen cuáles son sus nombres, los nombres de sus padres, los nombres de sus abuelos y los nombres de sus bisabuelos.

Con esa información, construyan sus árboles genealógicos en Java usando la clase **BinaryTree**.



Como un ejemplo, este sería el árbol genealógico del profesor



En la vida real, los árboles de ancestros son utilizados en Filogenética. Filogenética es la parte de la biología que se ocupa de determinar las relaciones evolutivas entre diferentes grupos de organismos

Si no conocen el nombre de uno de ellos, no lo escriban. No modelen hermanos ni tíos. Modelen en el método `main`, como hijos derechos, los hombres y, como hijos izquierdos, las mujeres. La raíz es “*Mauricio*”.



Escriban un método para la clase `BinaryTree` que calcule quién es la abuela materna de una persona.



Nota: Consideren que los ancestros paternos van a la derecha y los ancestros maternos a la izquierda.

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

2. Simulacro de Maratón de Programación sin documentación HTML, en GitHub, en la carpeta `ejercicioEnLinea`

2 Simulacro de Maratón de Programación: sin documentación, en GitHub, dentro de ejercicioEnLinea



Vean Guía
numeral 3.3



No se requiere
documentación para
los ejercicios en línea



Utilicen Java,
C++ o Python



No se reciben archivos
en **.PDF ni .TXT**



Código del
ejercicio en línea
en **GitHub**. Vean
Guía en numeral
4.24

! **Nota:** Si toman la respuesta de alguna fuente, deben **referenciar** según el **tipo de cita**. Vean *Guía* en **numerales 4.16 y 4.17**

2.1 Resuelvan el siguiente ejercicio usando árboles binarios:

Un árbol de búsqueda binario (o árbol binario de búsqueda) es un árbol binario que satisface las siguientes propiedades:

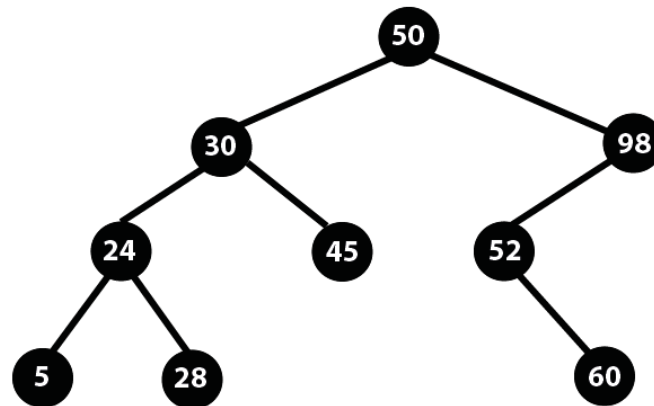
- El subárbol izquierdo de un nodo contiene sólo nodos con valores menores al valor del nodo.
- El subárbol derecho de un nodo contiene sólo nodos con valores mayores al valor del nodo.
- Tanto el subárbol izquierdo como el subárbol derecho tienen que ser también árboles de búsqueda binarios.

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



Ejemplo de un árbol de búsqueda binario

El recorrido en pre-orden (Raíz – Izquierda - Derecha) imprime los nodos visitando la raíz del árbol, luego recorriendo el subárbol izquierdo y luego recorriendo el subárbol derecho.

El recorrido en pos-orden (Izquierda – Derecha - Raíz) imprime el subárbol izquierdo primero, luego el subárbol derecho y por último la raíz. Por ejemplo, los resultados del recorrido en pre-orden y pos-orden del árbol de búsqueda binario mostrado en la Figura 1 son:

• **Pre-orden:** 50 30 24 5 28 45 98 52 60

• **Pos-orden:** 5 28 24 45 30 60 52 98 50



Dado el recorrido en pre-orden de un árbol de búsqueda binario, su tarea es encontrar el recorrido en pos-orden del árbol.



Entrada

El valor de cada nodo del árbol de búsqueda binario se dará de acuerdo a un recorrido en pre-orden. Cada nodo tiene un valor que es un entero positivo menor que 10^6 . Todos los valores se dan en líneas separadas (un entero por línea). Ud. puede asumir que un árbol de búsqueda binario no contiene más de 10 000 nodos y que no hay nodos duplicados (es decir, el valor de cada nodo es único en el árbol de búsqueda binario dado).

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



Salida

La salida contiene el resultado del recorrido en pos-orden del árbol de búsqueda binario dado en la entrada. Impriman un valor por línea.



Ejemplo de entrada

50
30
24
5
28

45
98
52
60



Ejemplo de salida

5
28
24
45
30
60
52
98
50

2.2 [Opc] Resuelvan el siguiente problema <http://bit.ly/2iNuHaa>

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

3. Simulacro de preguntas de sustentación de Proyecto en la carpeta informe

3 Simulacro de preguntas de sustentación de Proyecto: En la carpeta informe



Vean **Guía** numeral 3.4



Exporten y entreguen informe de laboratorio en **PDF, en español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**

Ejercicios sobre el Simulacro del Proyecto

3.1 Expliquen qué estructura de datos utilizaron para calcular las colisiones entre abejas, por qué la eligieron y qué complejidad tiene el algoritmo que, utilizando dicha estructura de datos, calcula las colisiones.

3.2 **[Opc]** ¿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?

Ejercicios sobre el simulacro de maratón de programación

3.3 **[Opc]** Expliquen con sus propias palabras cómo funciona la implementación del ejercicio 2.1 y opcionalmente el 2.2

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



Calculen la complejidad del ejercicio realizado en el numeral 2.1 y opcionalmente el 2.2.



Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.3



Ejemplos de su respuesta:

“n es el número de elementos del arreglo”,

“V es el número de vértices del grafo”,

“n es el número de filas de la matriz y m el número de columnas”.

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



4. Simulacro de parcial en informe PDF

4 Simulacro de Parcial en el informe PDF



Para este simulacro, agreguen ***sus respuestas*** en el informe PDF.



El día del Parcial no tendrán computador, JAVA o acceso a internet.

- 4.1** En los videojuegos, las tablas de hash se usan para guardar la información de las armas; por ejemplo, su ataque, su velocidad y su duración. Considera la siguiente función de hash para cadenas de caracteres. La constante `TABLE_SIZE` representa el tamaño máximo del arreglo con el que se representa internamente la tabla de hash.

```
private int funcionHash(String k){
    return ((int) k.charAt(0)) % TABLE_SIZE;
```

- ▶ ¿Qué problema presenta esta función hash? Las cadenas...
- (a) que terminan con la misma letra colisionan
 - (b) que inician con la misma letra colisionan
 - (c) cuya suma de los caracteres es la misma colisionan
 - (d) que tienen los mismos caracteres colisionan
- ▶ ¿Cuál es la complejidad asintótica, en el peor de los casos, de `funcionHash(k)`? Donde n es la longitud de la cadena k .
- (a) $O(n)$
 - (b) $O(n^2)$
 - (c) $O(n \log n)$
 - (d) $O(1)$

- 4.2** Según Andrés Mejía –egresado de Eafit; reconocido por haber trabajado en *Google* y *Facebook*; y, actualmente, ingeniero de software en *Riot Games*–, el principal reto que uno afronta al entrar a trabajar a una gran compañía –como *Riot Games*– es cómo entender el código que han hecho otras personas, entender su complejidad asintótica para el peor de los casos y poder realizar mejoras. El siguiente algoritmo es de vital importancia para empresas –como *Oracle* y *Microsoft*– porque se utiliza dentro de los compiladores de lenguajes orientados a objetos como Java y C#. Por si fuera

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

poco, este ejercicio es común en entrevistas para grandes empresas, según el portal *LeetCode*. Imagina que llegas nuevo a una de estas empresas, y debes entender y modificar este código. Lo único que sabemos es que `left_mis` es el resultado de `mystery` del árbol izquierdo y `right_mis` es el resultado de `mystery` del árbol derecho.

Si trabajas en Java, revisa este código:

```
public class BinaryTree {
    private Node root;
    Node mystery(int n1, int n2) {
        return mystery(root, n1, n2);
    }

    Node mystery(Node node, int n1, int n2) {
        if (node == null) return null;

        if (node.data == n1 || node.data == n2)
            return node;

        Node left_mis = mystery(node.left, n1, n2);
        Node right_mis = mystery(node.right, n1, n2);

        if (left_mis != null && right_mis != null)
            return node;

        return (left_mis != null) ? left_mis :
            right_mis;
    }
}
```

Si trabajas en Python, revisa este código:

```
def mystery(root , n1 , n2):
    if root is None:
        return None

    if root.key == n1 or root.key == n2:
        return root

    left_mis = mystery(root.left , n1 , n2)
    right_mis = mystery(root.right , n1 , n2)

    if left_mis and right_mis:
        return root

    return left_mis if left_mis is not None else
        right_mis
```

1. ¿Qué retorna la función `mystery`? Retorna el nodo.....
2. ¿Cuál es la complejidad **asintótica**, en el peor de los casos, del algoritmo anterior o la ecuación de recurrencia para el peor caso?
3. En una entrevista de Oracle, nos dicen que es un *árbol binario de búsqueda* (BST). ¿Cómo se puede hacer más eficiente el algoritmo `mystery` sabiendo que el árbol es un BST? Se puede.....

4.3 El siguiente problema es muy común en entrevistas de *Goldman Sachs* según el portal *Geeks for Geeks*. Dados dos arreglos *arr1* y *arr2*, de igual tamaño, la tarea es encontrar si los dos arreglos son iguales o no. Se dice que dos arreglos son iguales si ambos contienen el mismo conjunto de elementos, aunque el orden (o permutación) de los elementos puede ser diferente. Si hay repeticiones, entonces las ocurrencias de los elementos repetidos también deben ser iguales para que dos arreglos sean iguales. Las aplicaciones de este problema –en el sector bancario y financiero– son muy amplias, por la gran cantidad de transacciones.

Si trabajas en Java, revisa este código:

```

import java.util.*;
static boolean areEqual(int arr1[], int arr2
    []) {
    int n = arr1.length;
    int m = arr2.length;
    if (n != m)
        return false;
    Map<Integer, Integer> map = new HashMap<
        Integer, Integer>();
    int count = 0;
    for (int i = 0; i < n; i++) {
        if (map.get(arr1[i]) == null)
            map.put(arr1[i], 1);
        else {
            count = map.get(arr1[i]);
            count++;
            map.put(arr1[i], count);
        }
    }
    for (int i = 0; i < n; i++) {
        if (!map.containsKey(arr2[i]))
            return false;
        if (map.get(arr2[i]) == 0)
            return false;
        count = map.get(arr2[i]);
        --count;
        map.put(arr2[i], count);
    }
    .....
}

```

Si trabajas en Python, revisa este código:

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

```

from collections import defaultdict
def areEqual(arr1, arr2, n, m):
    if (n != m):
        return False;
    count = defaultdict(int)
    for i in arr1:
        count[i] += 1
    for i in arr2:
        if (count[i] == 0):
            return False
        else:
            count[i] -= 1
    .....

```

1. Completa la última línea.....
2. ¿Cuál es la complejidad asintótica, en el peor de los casos, de areEqual? Donde n es el número de elementos del primer arreglo y m el número de elementos del segundo arreglo. Es $O(\text{.....})$

4.4 Consideren la siguiente definición de árbol binario:

```

class Node {
    public Node left;
    public Node right;
    public String data;
    public Node(String d) {
        data = d;
    }
}

```

El siguiente algoritmo imprime todos los valores de un árbol en pre orden.

```

01 private void printAUX(Node node) {

```

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

```

02     if (node != null) {
03         System.out.println(node.data);
04         printAUX(node.left);
05         printAUX(node.right);
06     }
07 }
08 public boolean print() {
09     printAUX(root);
10 }

```

4.4.1 ¿Cuál ecuación de recurrencia que describe el número de instrucciones que ejecuta el algoritmo *print* en el peor de los casos?

La variable n representa el número de elementos del árbol.

▶ De acuerdo a lo anterior, elijan la respuesta que consideren acertada:

- a) $T(n) = T(n-1) + C$, que es $O(n)$
- b) $T(n) = 2 \cdot T(n-1) + C$, que es $O(2^n)$
- c) $T(n) = 2 \cdot T(n/2) + C$, que es $O(n)$
- d) $T(n) = T(n/2) + C$, que es $O(\log_2 n)$
- e) $T(n) = T(n+1) + C$, que es infinito

4.4.2 ¿Cuál ecuación de recurrencia que describe el número de instrucciones que ejecuta el algoritmo *print* en el peor de los casos?

▶ Elijan la respuesta que consideren acertada:

- a) $O(n)$
- b) $O(n^2)$
- c) $(\log n)$
- d) $O(n.m)$
- e) $O(1)$

4.4.3 ¿Cuál es la salida del algoritmo *print* para el siguiente árbol?

Tengan en cuenta que la raíz es Wilkenson. Después de hacer la rotación, usted entenderá que realmente las mujeres van a la izquierda y los hombres a la derecha.

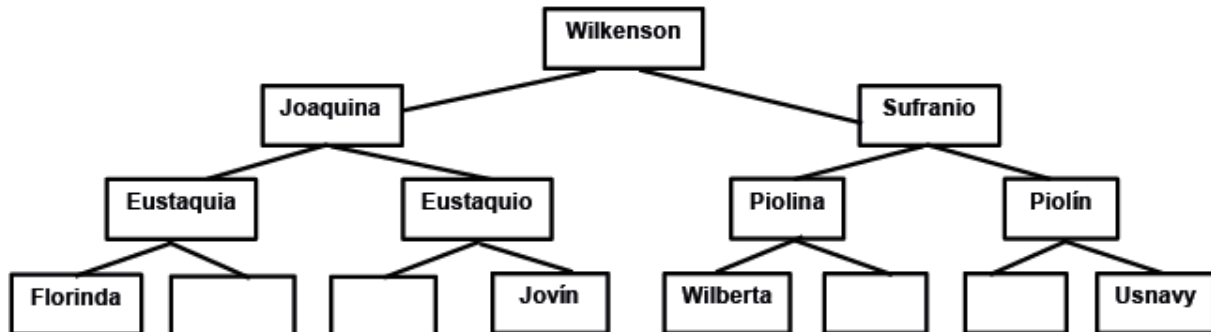
PhD. Mauricio Toro Bermúdez

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

Además, en un árbol genealógico, para algunas personas, no se conoce la mamá o el papá.



▶ Elijan la respuesta que consideren acertada:

- a) Wilkenson, Sufranio, Piolín, Usnavy, Piolina, Wilberta, Joaquina, Eustaquio, Florinda, Eustaquia, Yovín
- b) Sufranio, Piolina, Wilberta, Piolín, Usnavy, Joaquina, Estaquia, Florinda, Wilkenson, Yovín, Eustaquio
- c) Wilkenson, Yovín, Eustaquio, Sufranio, Piolina, Wilberta, Piolín, Usnavy, Joaquina, Estaquia, Florinda
- d) Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolín, Usnavy
- e) Sufranio, Piolina, Wilberta, Piolín, Usnavy, Florinda, Wilkenson, Yovín, Eustaquio, Joaquina, Estaquia

4.4.4 ¿Qué modificación hay que hacer algoritmo *print* para que arroje la siguiente respuesta para el árbol anterior?:

Usnavy, Piolín, Wilberta, Piolina, Sufranio, Florinda, Eustaquio, Yovín, Eustaquia, Joaquina, Wilkenson.

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

Tengan en cuenta que la raíz es Wilkenson. Como Wilkenson es la raíz, para poder entender el árbol, recomendamos rotarlo 180 grados. Después de hacer la rotación, usted entenderá que realmente las mujeres van a la izquierda y los hombres a la derecha

Además, en un árbol genealógico, para algunas personas, no se conoce la mamá o el papá.



Elijan la respuesta que consideren acertada:

- a) Cambiar el orden de las líneas 03, 04 y 05 por 05, 04, 03
- b) Cambiar el orden de las líneas 03, 04 y 05 por 04, 05, 03
- c) Cambiar el orden de las líneas 03, 04 y 05 por 03, 05, 04
- d) Cambiar el orden de las líneas 03, 04 por 06, 07
- e) Intercambiar la línea 03 y la línea 04

4.5

Luis escribió un programa para insertar un número en un árbol binario de búsqueda. En dicho árbol, él quiere tener los números menores o iguales a la raíz a la derecha y los mayores a la raíz a la izquierda.

Ayúdele a completar su código. El algoritmo recibe la raíz de un árbol p y un número a insertar `toInsert`, y retorna la raíz del árbol con el elemento insertado donde corresponde.

```
01 private Node insert(Node p,int toInsert){
02   if (p == null)
03     return new Node(toInsert);
04   if (.....)
05     return p;
06   if (.....)
07     p.left = insert(p.left, toInsert);
08   else
09     p.right = insert(p.right, toInsert);
10   return p;
11 }
```



Completen los espacios en blanco, así:

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

a) Completen, por favor, la línea 4 con la condición que corresponde

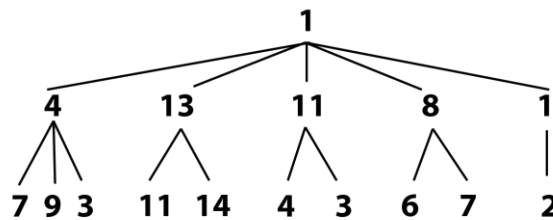
b) Completen, por favor, la línea 6 con la condición que corresponde

4.6 [Opc] Dado un árbol *n-ario*, un camino desde la raíz a cualquiera de sus hojas se considera **simple** si la suma de sus elementos pares es igual a la suma de sus elementos impares.

Por ejemplo, un camino desde la raíz hasta una hoja con los elementos [1, 2, 1] es **simple**, pero el camino desde la raíz hasta una hoja [1, 3, 1] **no es simple**.

Su tarea es determinar cuántos caminos **simples** hay en un árbol. La implementación de un nodo *n-ario* es la siguiente:

```
class NNode{
    int val; //Valor en el nodo actual.
    //Hijos del nodo actual.
    LinkedList<NNode> hijos;
}
```



▶ **4.6.1** ¿Cuántos caminos **simples** hay desde una raíz hasta una hoja en el árbol anterior?

- a. 1
- b. 2
- c. 3
- d. 4

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

El siguiente código nos ayuda a determinar el número de caminos simples en un árbol, pero faltan algunas líneas. Por favor, complétenlas.

```

01 public int cuantosSimples(NNodo raiz, int suma){
02     //Arbol vacio
03     if(raiz == null)
04         _____;
05     //Hoja
06     if(raiz.hijos.size() _____) // Si suma es 0
07         return (suma == 0) ? 1 : 0; // Retorne 1, sino, 0
08     int total = 0;
09     for(NNodo n: raiz.hijos)
10         if(n.val % 2 == 0) // Par
11             total += cuantosSimples(n, suma + n.val);
12         else // Impar
13             total += cuantosSimples(n, suma - n.val);
15     return total;
16 }
17
18 public int cuantosSimples(NNodo raiz){
19     int val = (raiz.val % 2 == 0) ?
20         raiz.val : -raiz.val;
21     return cuantosSimples(raiz, val);
22 }

```



Completen los espacios en blanco, así:

4. 6.2 Completen la línea 4

4. 6.3 Completen la línea 6

4.7 [Opc] Los siguientes algoritmos son el recorrido en in orden y pos orden. En la vida real, estos recorridos son de utilidad para hacer procesamiento del lenguaje natural.

```
void InOrden(Node node) {
```

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

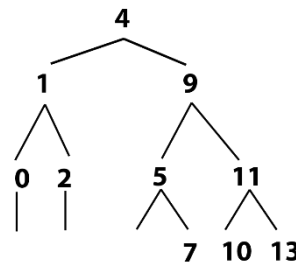
```

if (node != null){
    InOrden(node.left);
    System.out.println(node.data);
    InOrden(node.right);
} }

void PosOrden(Node node){
    if (node != null){
        PosOrden(node.left);
        PosOrden(node.right);
        System.out.println(node.data);
    } }

```

Consideren el recorrido in orden y pos orden de un **Árbol binario de búsqueda** con los siguientes elementos 4, 9, 1, 5, 7, 11, 13, 2, 0, 10. Ahora, la impresión del recorrido in-orden nos devolverá los elementos, y, la impresión del recorrido pos-orden devolverá.



4.7.1 ¿Cuál es la impresión pos-orden?



Eliján la respuesta que consideren acertada:

- a) 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
- b) 0, 1, 2, 4, 5, 7, 9, 10, 11, 13
- c) 0, 2, 1, 7, 10, 5, 13, 11, 9, 4
- d) 4, 1, 0, 2, 9, 5, 7, 11, 13, 10

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

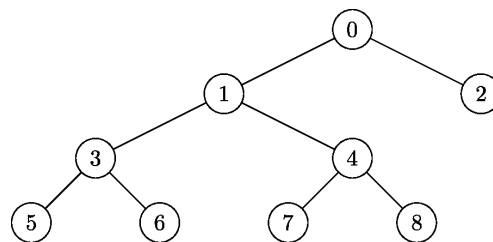
ESTRUCTURA DE DATOS 1
Código ST0245

4.7.2 Supongan que hacemos la comparación elemento a elemento del recorrido en in-orden y el recorrido en pos-orden. ¿Cuántos elementos aparecen en la misma posición en ambos recorridos? Como uno ejemplo, si un recorrido es 1,3,4,2 y el otro es 2,3,4,1, la respuesta es 2 porque el 3 y el 4 aparecen en la misma posición en ambos.

▶ **Elijan la respuesta que consideren acertada:**

- a) 7
- b) 2
- c) 5
- d) 8

! **Nota: Consideren el siguiente árbol para los ejercicios 8,9 y 10**



▶ **Elijan la respuesta que consideren acertada:**

- a) $O(n^3)$
- b) $O(n^2)$,
- c) $O(\log n)$
- d) $O(n)$

4.8 [Opc] Sean A y B las salidas de los recorridos pre-orden y pos-orden del árbol binario anterior, respectivamente. Determinen el número de elementos para los cuales se cumple que para $1 \leq i \leq 8$.

▶ **Elijan la respuesta que consideren acertada:**

- a) 3

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

- b) 2
- c) 4
- d) 0

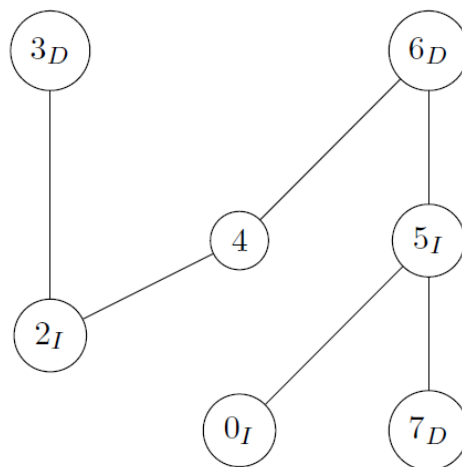
4.9 [Opc] ¿Cuál es la salida del recorrido in-orden del árbol binario anterior?



Elijan la respuesta que consideren acertada:

- a) 5, 3, 6, 1, 7, 4, 8, 0, 2
- b) 0, 1, 3, 5, 6, 4, 7, 8, 2
- c) 5, 6, 3, 7, 8, 4, 1, 2, 0
- d) 5, 6, 3, 1, 7, 4, 8, 0, 2

4.10 [Opc] El sub-índice D indica que el nodo es un hijo derecho con respecto a su padre. El sub-índice I indica que es un hijo izquierdo con respecto a su padre



4.10.1 ¿Cuál es la salida del recorrido en *in-orden* del árbol anterior, tomando el nodo 4 como el nodo raíz?

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



Elijan la respuesta que consideren acertada:

- a) 3, 2, 0, 7, 5, 6, 4
- b) 2, 3, 4, 0, 5, 7, 6
- c) 4, 2, 3, 6, 0, 5, 7
- d) 4, 7, 3, 6, 2, 0, 5

4.10.2 Asuma que el nodo 4 es la raíz del árbol binario anterior. Asuma que la salida del recorrido *pre-orden* es $\{a_1, a_2, a_3, \dots, a_n\}$ y la salida del recorrido *in-orden* es $\{b_1, b_2, b_3, \dots, b_n\}$.

¿Cuál es el primer valor de i para el cual $a_i = b_i$? Note que la i empieza en 1.

- a) 5
- b) 4
- c) 3
- d) 3



Por ejemplo, asuman que las salidas son $a = \{4, 6, 5, 0, 7, 2, 3\}$ y $b = \{6, 4, 7, 0, 5, 2, 3\}$. La respuesta sería 4 porque $a_4 = b_4 = 0$ e $i=4$ es el primer valor para el que $a_i = b_i$.

4.10.3 ¿Es un **árbol binario de búsqueda** el árbol anterior?



Elijan la respuesta que consideren acertada:

- a) Sí
- b) No

4.11

[Opc] La siguiente es una implementación de árbol n-ario.

```
1 class Nodo {
2     public int id;
3     public int valor;
4     public LinkedList<Nodo> hijos;
5 }
```

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

Hoy, Liko y Kefa, en clase de Estructura de Datos 1, aprendieron que con los árboles n -arios se pueden hacer muchísimas operaciones, tales como encontrar el tamaño de un sub-árbol, calcular la suma de los elementos en un sub-árbol, etc. Muchas de estas operaciones son realmente fáciles para Liko; sin embargo, hay una –en especial– difícil para Kefa. Para Kefa es difícil escribir un programa que calcule la suma de los elementos de los sub-árboles. Liko ha escrito un programa para calcular la suma de los elementos en los sub-árboles para enseñarle a Kefa e intencionalmente ha omitido una parte para asegurarse que, si Kefa entiende el código, él sea capaz de encontrar el error. ¿Podrías ayudarles a Kefa a encontrar el error, por favor? Para propósitos de este ejercicio, Liko se ha asegurado que el árbol dado tiene exactamente n nodos, y que al método suma se le pasa la raíz del árbol y un arreglo con n ceros.

```

1 void suma(Nodo raíz, int[] suma){
2     if(raíz.hijos == null || raíz.hijos.size() == 0){
3         suma[raíz.id] = raíz.valor;
4         return;
5     }
6     suma[raíz.id] = raíz.valor;
7
8     for(Nodo e: raíz.hijos){
9         suma(e, suma);
10        suma[_____] = suma[e.id] + suma[raíz.id];
11    }
12 }
```

4.11.1 Completa, por favor, la línea 10 _____

4.11.2 ¿Cuál es la complejidad asintótica, en el peor de los casos, del código anterior?

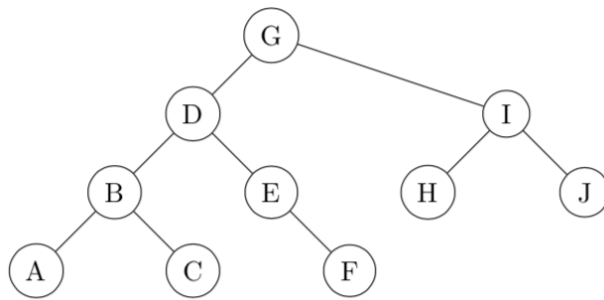
- a) $T(n) = T(n - 1) + c$, que es $O(n)$
- b) $T(n) = 4T(n/2) + c$, que es $O(n^2)$
- c) $T(n) = 2T(n - 1) + c$, que es $O(2^n)$
- d) $T(n) = nT(n - 1) + c$, que es $O(n!)$

4.12 [Opc] Dada la siguiente lista de enteros, ubica los números en el árbol, de tal manera que el árbol final sea un árbol binario de búsqueda:
7,9,10,8,4,5,6,2,1,3

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



4.12.1 ¿Cuáles elementos van en cada letra si se insertan en el orden dado anteriormente?

- i) $A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8, I = 9, J = 10.$
- ii) $A = 4, B = 5, C = 6, D = 7, E = 8, F = 9, G = 1, H = 10, I = 2, J = 3.$
- iii) $A = 7, B = 8, C = 9, D = 10, E = 1, F = 2, G = 3, H = 4, I = 5, J = 6.$
- iv) $A = 7, B = 8, C = 9, D = 10, E = 1, F = 2, G = 3, H = 4, I = 5, J = 6.$

4.12.2 ¿Cuál sería el recorrido pre-orden del árbol anterior, asumiendo que siempre se visita primero la hoja de la izquierda?

- a) $G, D, B, A, C, E, F, I, H, J$
- b) $A, B, C, D, E, F, G, H, I, J$
- c) $B, C, D, E, F, G, H, I, J, A$
- d) $A, C, B, D, I, J, H, E, F, G$

4.12.3 ¿Cuál es la complejidad asintótica, en el peor de los casos, de encontrar una clave (número) en un árbol binario de búsqueda de n nodos?

- a) $O(n)$
- b) $O(\log(n))$
- c) $O(n^2)$
- d) $O(1)$

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

5. [Opcional]

Lecturas Recomendadas

5 [Opc] Lecturas recomendadas



Vean *Guía* en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF**, en **español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



"El ejercicio de una profesión requiere la adquisición de competencias que se sustentan en procesos comunicativos. Así cuando se entrevista a un ingeniero recién egresado para un empleo, una buena parte de sus posibilidades radica en su capacidad de comunicación; pero se ha observado que esta es una de sus principales debilidades..." *Tomado de <http://bit.ly/2qJKzJD>*



Vean *Guía* en **numerales 3.5 y 4.20**



Lean a "**Robert Lafore, Data Structures and Algorithms in Java (2nd edition), 2002. Chapter 8: Binary Trees.**", y sumen puntos adicionales, así:



Hagan un mapa conceptual con los principales elementos teóricos.

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



Nota: Si desean una lectura adicional en inglés, consideren la siguiente:
“Narasimha Karumanchi, Data Structures and Algorithms made easy in Java, (2nd edition), 2011. Section 6.9 BST”, que pueden encontrarla en biblioteca

6. [Opcional] Trabajo en Equipo y Progreso Gradual

6 [Opc] Trabajo en Equipo y Progreso Gradual



Vean Guía en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF**, en **español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



El trabajo en equipo es una exigencia del mercado. "Mientras algunos medios retratan la programación como un trabajo solitario, pero requiere mucha comunicación y trabajo grupal. Si trabajas para una compañía, serás parte de un equipo de desarrollo y esperarán que te comuniques bien con otras personas" Tomado de <http://bit.ly/1B6hUDp>



Vean Guía en **numerales 3.6, 4.21, 4.22, 4.23**

6.1



Entreguen copia de todas las actas de reunión usando el **tablero Kanban**, con fecha, hora e integrantes que participaron

6.2



Entreguen el reporte de **git** con los cambios en el código y quién hizo cada cambio, con fecha, hora e integrantes que participaron

6.3



Entreguen el reporte de cambios del informe de laboratorio que se genera **Google docs** o herramientas similares



Nota: Estas respuestas también deben incluirlas en el informe PDF

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

7. [Opcional]

Laboratorio en Inglés con plantilla en Inglés

7 [Opc] Laboratorio en inglés



Vean Guía en **numeral 3.5 y 4.20**



Exportar y entregar informe de laboratorio en **PDF, en español o Inglés**



Si hacen el **informe en español**, usen la **plantilla en español**



No apliquen **Normas Icontec** para esto

Si hacen **el informe en inglés**, usen a **plantilla en inglés**



El inglés es un idioma importante en la Ingeniería de Sistemas porque la mayoría de los avances en tecnología se publican en este idioma y la traducción, usualmente se demora un tiempo.

Adicionalmente, dominar el inglés permite conseguir trabajos en el exterior que son muy bien remunerados. Tomado de goo.gl/4s3LmZ

▶ **Entreguen el código y el informe en inglés.**

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

Ayudas para resolver los Ejercicios

Ayudas para el Ejercicio 1.1.....	<u>Pág. 38</u>
Ayudas para el Ejercicio 1.3.....	<u>Pág. 38</u>
Ayudas para el Ejercicio 2.1.....	<u>Pág. 39</u>
Ayudas para el Ejercicio 2.2.....	<u>Pág. 39</u>
Ayudas para el Ejercicio 3.2.....	<u>Pág. 39</u>
Ayudas para el Ejercicio 3.4.....	<u>Pág. 39</u>
Ayudas para el Ejercicio 4.....	<u>Pág. 40</u>
Ayudas para el Ejercicio 5.....	<u>Pág. 40</u>
Ayudas para el Ejercicio 6.1.....	<u>Pág. 40</u>
Ayudas para el Ejercicio 6.2.....	<u>Pág. 40</u>
Ayudas para el Ejercicio 6.3.....	<u>Pág. 40</u>



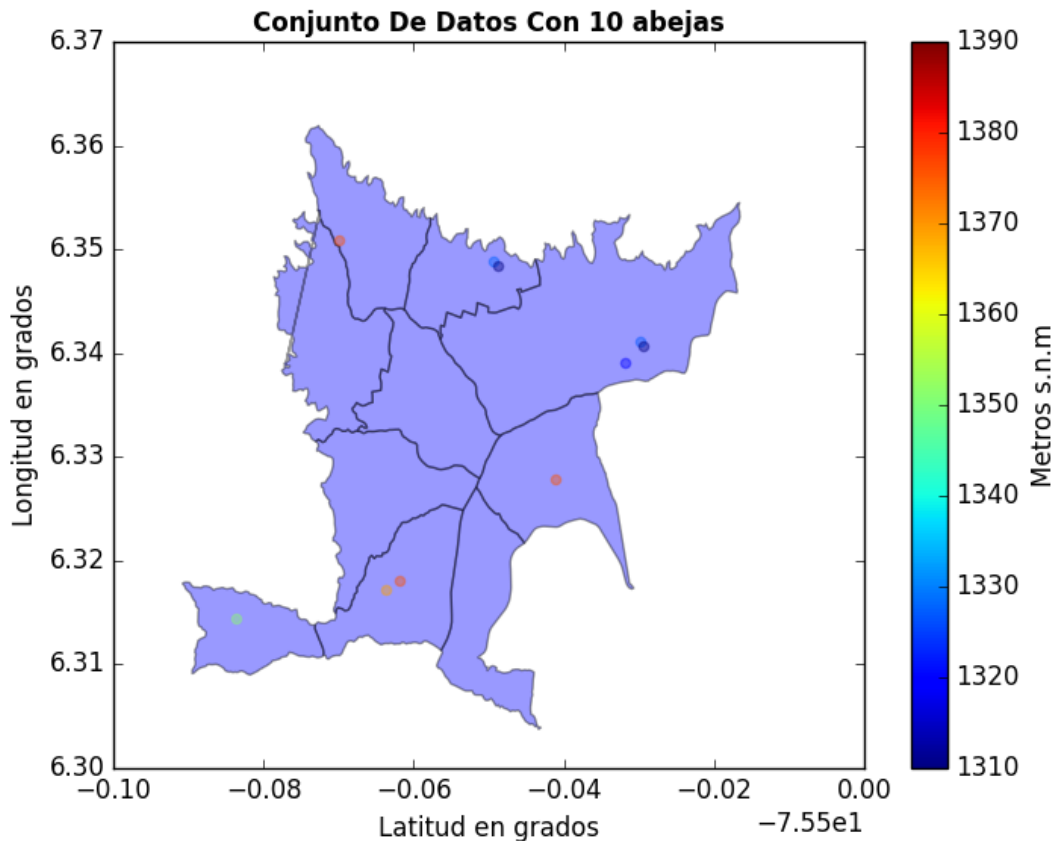
Ayudas para el Ejercicio 1.1



Pista: Si deciden hacer la documentación, consulten la *Guía en numeral 4.1*



Como un ejemplo, en la Gráfica 2 se muestra un mapa de Bello, Antioquia con 10 abejas robóticas. En la Tabla 1 se muestran las coordenadas geodésicas de las abejas que están en la Gráfica 2. En la Tabla 2 se muestra la salida esperada del algoritmo.



Gráfica 2. Georreferenciación de las abejas robóticas en el mapa del municipio de Bello, Antioquia.

-75.5618619706,6.31811259895,1375.92
 -75.5699054633,6.35084148782,1375.89
 -75.5636848415,6.31724585117,1368.81

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

```
-75.5836265641,6.31450564533,1351.39
-75.5318618084,6.33905462111,1318.96
-75.5298058637,6.34118508844,1327.07
-75.5492944209,6.34887049624,1327.65
-75.5410449586,6.32792514147,1375.53
-75.5488444205,6.34842049579,1307.65
-75.5293558632,6.34073508799,1307.07
```

Tabla 1. Coordenadas geodésicas de las abejas. La primera componente representa la latitud en grados, la segunda la longitud en grados y la tercera la altura sobre el nivel del mar en metros. Cada línea es una abeja.

```
-75.5492944209, 6.34887049624, 1327.64
-75.5488444204, 6.34842049578, 1307.64
-75.5298058637, 6.34118508844, 1327.07
-75.5293558632, 6.34073508799, 1307.07
```

Tabla 2. Coordenadas geodésicas de las abejas que se encuentran a 100 metros de alguna otra. La primera componente representa la latitud en grados, la segunda la longitud en grados y la tercera la altura sobre el nivel del mar en metros. Cada línea es una abeja.

En la carpeta de conjuntos de datos (*datasets/abejas*) encontrarán archivos de texto con la información de la Tabla 1, al igual que la imagen que se muestra en la Gráfica 1, para diferentes cantidades de abejas.



Ejemplos de estructuras de datos basadas en árboles y tablas de hash comúnmente utilizadas para detección de colisiones.

<https://www.gamedev.net/articles/programming/general-and-gameplay-programming/spatial-hashing-r2697/>

<http://www.azurefromthetrenches.com/introductory-guide-to-aabb-tree-collision-detection/>

<https://en.wikipedia.org/wiki/Quadtree>

<http://www.randygaul.net/2013/08/06/dynamic-aabb-tree/>

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



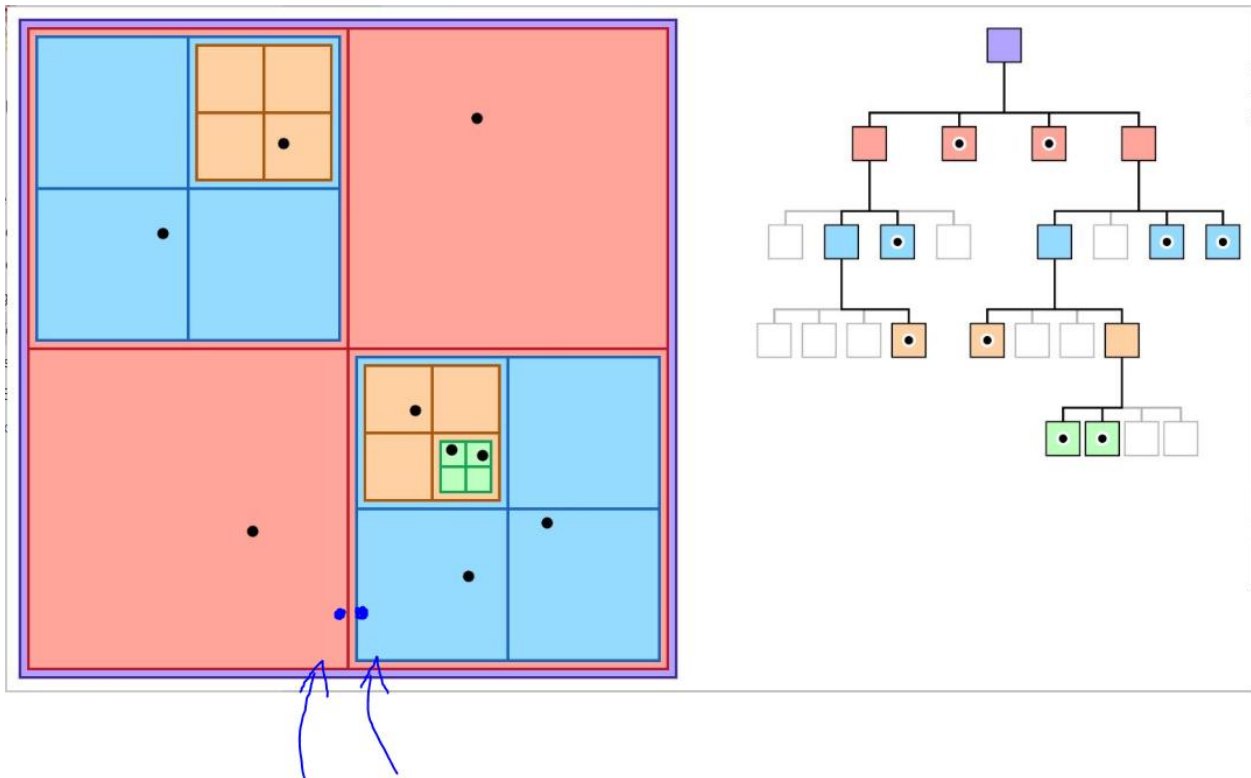
ESTRUCTURA DE DATOS 1

Código ST0245



A continuación, algunas alternativas de solución.

Un quadtree es un árbol que se forma dividiendo el espacio en 4. Si hay una abeja, no más divisiones. Si hay 2 o más abejas, vuelves a dividir el espacio en 4. Si hay una abeja, descanso. Sino, vuelvo a dividir el espacio en 4, y así sucesivamente. Eso permite que yo no tenga que comparar millones de abejas de la esquina superior izquierda con millones de abejas de la esquina inferior derecha. Simplemente, si están en cuadrados diferentes, no hay colisiones entre esas abejas y ya.



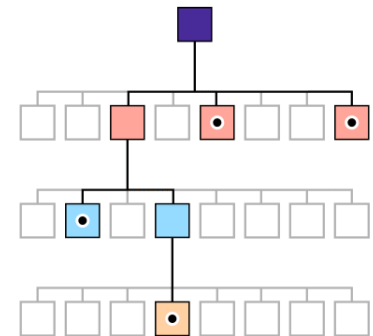
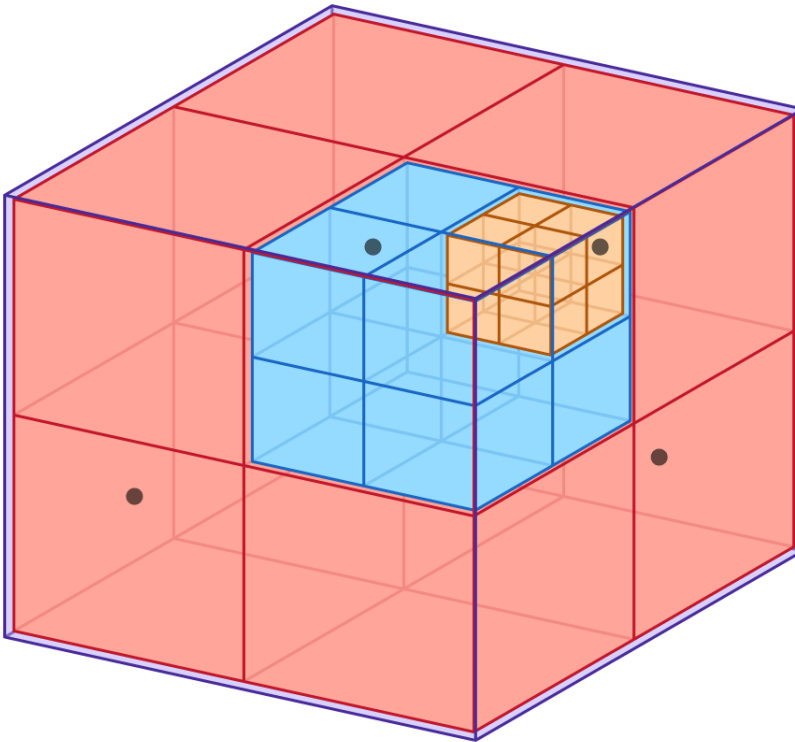
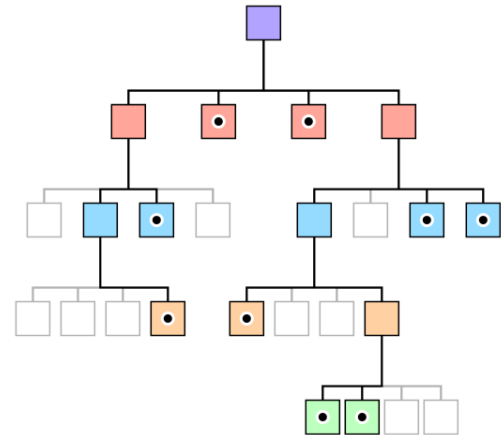
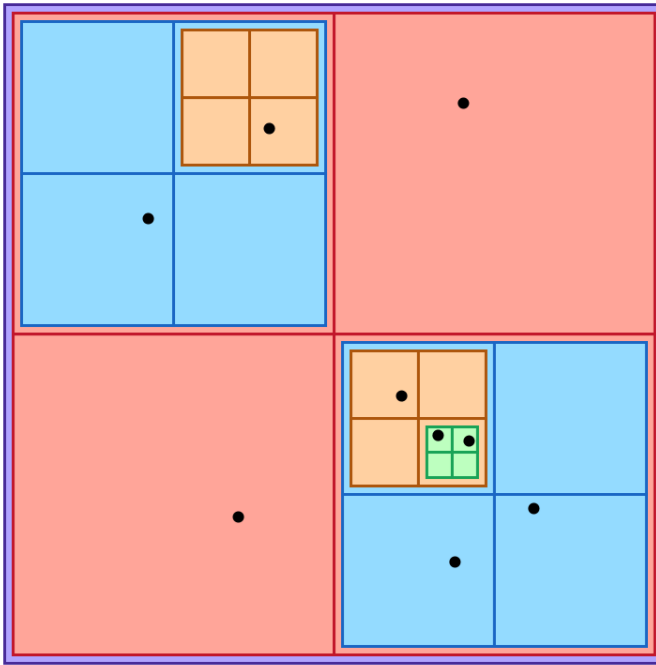
Bueno, un octree es como un quadtree, pero ahora es un cubo y cada cubo se divide en 8 subcubos, y así sucesivamente. Esto por es en 3D, como realmente son la mayoría de los videojuegos...y en nuestro caso las abejas también.

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

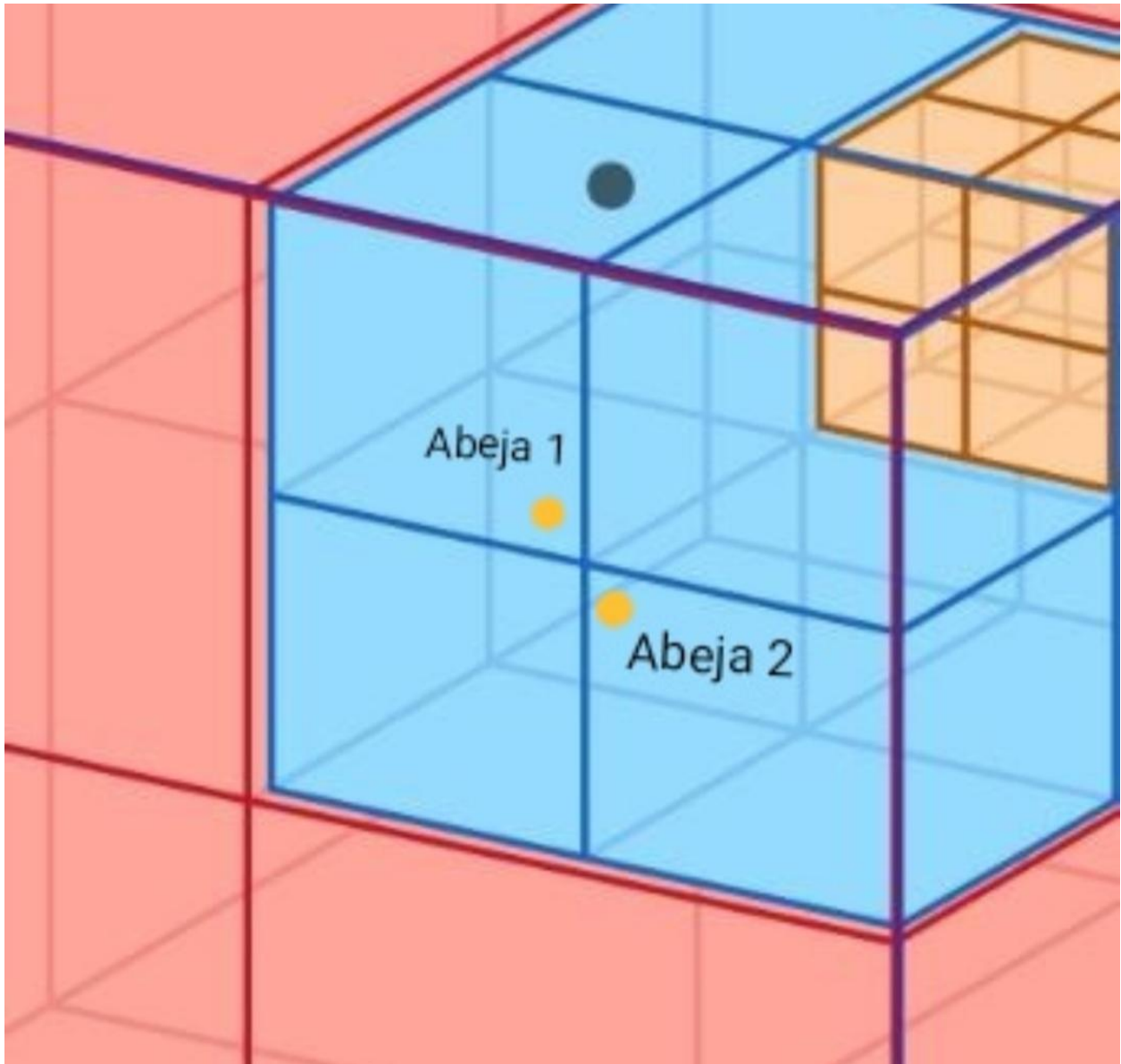
Código ST0245



No obstante, hay un problema y es qué pasa si hay dos abejas que están a menos de 100 metros pero en cuadrantes diferentes.

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



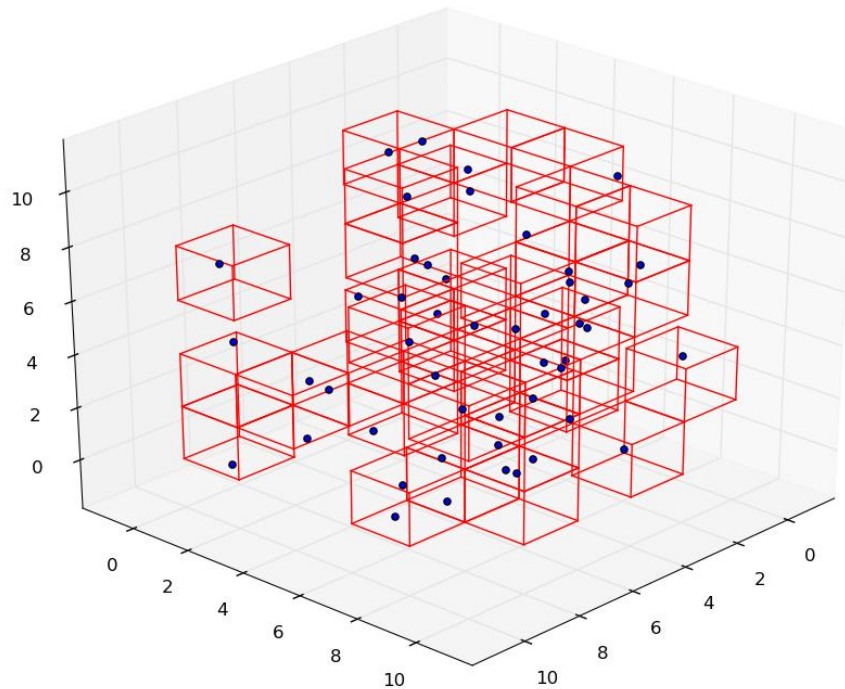
Otra opción es hacer una tabla de hash espacial en 3D, de esta forma:

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



Ayudas para el Ejercicio 1.3



Como un ejemplo, en el árbol genealógico de Wilkenson, Eustaquia es la abuela materna de Wilkenson. Otro ejemplo, Wilberta es la abuela materna de Sufranio y no se conoce la abuela materna de Eustaquia



Pista 1: Utilicen el siguiente estereotipo para el método:

```
public String getGrandMothersName(String name)
```



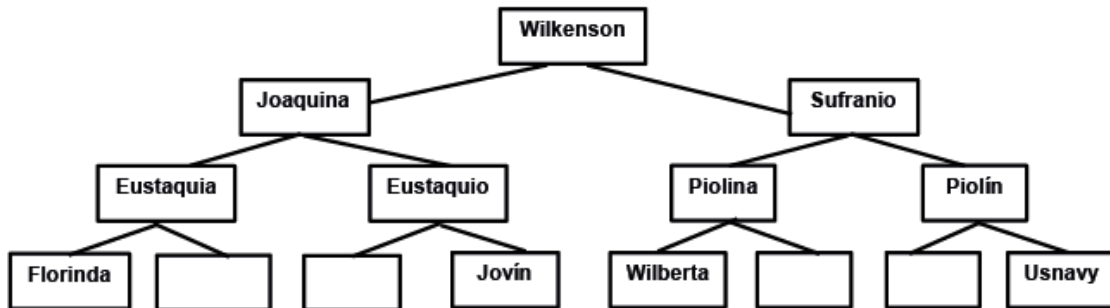
PhD. Mauricio Toro Bermúdez

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

Pista 2: Retornen una cadena vacía si no se encuentra el nombre de la persona *name* o la abuela materna de *name*. A continuación, un ejemplo de un árbol genealógico



Ayudas para el Ejercicio 2.1



Pista 1: Lean sobre <http://bit.ly/1V5RqKL>



Pista 2: No es posible reconstruir un árbol binario, en general, con un el recorrido pre-orden porque habría muchas posibles respuestas. Para poder hacer este ejercicio hay que recordar que es un árbol binario de búsqueda, es decir, donde los menores van a la izquierda y los mayores van a la derecha.



Ayudas para el Ejercicio 2.2



Pista: Use *búsqueda en profundidad (pre orden, in orden o pos orden)*. Es similar a lo que hacen en Lenguajes de Programación para construir el árbol sintáctico en una expresión aritmética con paréntesis.



Ayudas para el Ejercicio 3.2

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



Pista 1: Lean sobre <http://bit.ly/1V5RqKL>



Pista 2: Lean sobre <http://bit.ly/2i8DgM4>



Ayudas para el Ejercicio 3.4



Pista 1: Vean *Guía en numeral 4.11*



Pista 2: Vean *Guía en numeral 4.19*



Ayudas para el Ejercicio 4



Pista 1: Vean *Guía en numeral 4.18*



Ayudas para el Ejercicio 5



Pista 1: Para que hagan el mapa conceptual se recomiendan herramientas como las que encuentran en <https://cacoo.com/> o <https://www.mindmup.com/#m:new-a-1437527273469>



Ayudas para el Ejercicio 6.1

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



Pista 1: Vean Guía en numeral 4.21



Ayudas para el Ejercicio 6.2



Pista 1: Vean Guía en numeral 4.23



Ayudas para el Ejercicio 6.3



Pista 1: Vean Guía en numeral 4.22

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

¿Alguna inquietud?

CONTACTO

Docente Mauricio Toro Bermúdez

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

Correo: mtorobe@eafit.edu.co

Oficina: 19- 627

Agenden una cita dando clic en la pestaña

-*Semana*- de <http://bit.ly/2gzVg10>