

Laboratorio 04

Árboles binarios y Tablas de Hash

Maria Alejandra Vélez Clavijo
Universidad Eafit
Medellín, Colombia
mavelezc1@eafit.edu.co

Laura Katterine Zapata Rendón
Universidad Eafit
Medellín, Colombia
lkzapatar@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 El objetivo de este algoritmo es identificar aquellas abejas que se encuentren a 100 metros o menos de distancia de otra abeja, para así poder prevenir las colisiones entre ellas. Sea n : el número de abejas, la complejidad del algoritmo para calcular las colisiones es $O(n)$. En donde la totalidad de los métodos de las clases Octree son también $O(n)$, excepto la función hashing, esta es de $O(1)$.

Se tiene en cuenta que colisionaran aquellas abejas que estén a 100 o menos metros de distancia. Para calcular las colisiones de un grupo de abejas robóticas en 3D de acuerdo a su longitud (grados), latitud (grados) y altura (metros) se implementa un árbol octal cuyos nodos tienen cada uno 8 nodos hijos.

Para esto se implementan arreglos de 8 posiciones en donde cada posición almacenará una lista enlazada de abejas y el objetivo es dividir estas listas enlazadas en subárboles de 8 nodos recursivamente hasta que cada lista enlazada cuente con una sola abeja o hasta que se identifique que este subgrupo de abejas colisionará. Para saber esto último se tiene un método que calcula la diagonal de un cubo en 3D en donde una esquina será la posición con menor latitud, longitud y altura y la otra será la posición con mayor latitud, longitud y altura. Se debe tener en cuenta que para este cálculo se deben pasar los valores de longitud y latitud de grados a metros (multiplicando por 111111).

Luego, descartando que el rango de posiciones de este subgrupo de abejas es mayor a 100 metros, se procede a crear un nuevo arreglo con 8 posiciones, en donde en cada posición se almacenara una lista enlazada de abejas. Para saber en qué posición va cada abeja se usa la función hashing la cual recibe como parámetro la abeja que se quiere posicionar y un array de doubles donde están los valores medios de la longitud, la latitud y altura del subgrupo al cual pertenece la abeja. Los valores medios se calculan restando el máximo menos el mínimo, dividirlo entre 2 y luego ese resultado sumándolo al mínimo, así estaríamos obteniendo el valor que está en el centro del mínimo y el máximo.

La función hashing lo que hace es dividir ese espacio en 3 dimensiones que forma el subgrupo donde está la abeja y dividirlo en 8 sub-cubos de acuerdo al valor medio de cada

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

coordenada. Así el cubo “0” está conformado por todas las abejas cuya longitud, latitud y altura este debajo del valor medio de cada coordenada. Así se garantiza que cada abeja pertenezca a uno y solamente un cubo de los 8. Luego de identificar a cuál cubo pertenece la abeja, la función hashing retorna un entero de 0 a 7 el cual será la posición del arreglo al cual pertenece. Finalmente se tiene un método choque() que recibe la lista enlazada de aquellas abejas que colisionan e imprime las coordenadas de cada una de ellas.

3.2

3.3 Ejercicio 2.1

En el ejercicio 2.1 pasan a un arreglo de enteros los cuales van a ser insertados en un árbol. Para hacer dicho trabajo se tiene el método buildingTree() el cual permite iterar por cada elemento de ese arreglo e irlo insertando en el árbol utilizando el método insert(), este último permite insertar los elementos tomando como referencia la raíz (que será el primer elemento del arreglo), hace unas comparaciones para saber si el elemento a ingresar es menor, mayor o igual a la raíz y saber en qué posición del árbol debe ir.

Aparte de eso está el método posOrder, que nos va a permitir ver los elementos que antes ingresamos, mostrándonos el nodo de la izquierda, luego derecha y por último raíz.

Ejercicio 2.2

En el ejercicio 2.2 nos pasan un String el cual tiene una forma particular, pues este nos representa la estructura de un árbol según como estén ubicados los elementos respecto a paréntesis. Inicialmente se utiliza la clase suma a la cual le vamos a pasar el String y esta se va encargar de recorrerlo y de acuerdo a unas condiciones va ir insertando cada elemento del string en un árbol. Luego de que la clase suma nos permite crear el árbol, le pasamos la raíz de dicho árbol al método sumaElCamino el cual se va encargar de recorrer todos los posibles caminos del árbol, tomando rutas tanto en la izquierda como en la derecha, sumando los valores de los nodos y así mismo va verificando si la suma de estos coincide con el valor a comparar. Finalmente nos devuelve verdadero si pudo encontrar una ruta, la cual con la suma de sus caminos coincidió con el valor suma dado.

3.4 complejidad del ejercicio 2.1

// $T(n) = n \log n + n$

// $O(n \log n)$

complejidad del ejercicio 2.2

// $T(n, m) = m \cdot \log n$

// $O(m \log n)$

3.5

En el ejercicio 2.1 // n : total de elementos del árbol

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

En el ejercicio 2.2 // m: numero de subcadenas del string
// n: numero de elementos del árbol

3.6

4) Simulacro de Parcial

4.1

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

4.2 c) 3

4.3

- false
- a.data()
- a.izq, suma-a.data()
- a.der, suma-a.data()

4.4

- c) $T(n)=2.T(n/2)+C$
- a) $O(n)$
- d) Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolín, Usnavy
- a) Cambiar el orden de las líneas 03, 04 y 05 por 05, 04, 03

4.5

- toInsert==null
- toInsert>p

4.6

- d. 4
- return 0
- ==0

4.7

- a) 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
- b) 2
- para imprimir todos los nodos : d) $O(n)$
- para buscar un nodo sería c) $O(\log n)$ // Nota: no especificaron en la pregunta

4.8 b) 2

4.9

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

4.10 No hay

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

4.11

- b) 2, 3, 4, 0, 5, 7, 6
- a) 5
- b) No. No es un árbol binario de búsqueda porque en este caso, los números de la derecha no siempre son mayores que los de la raíz.

4.12

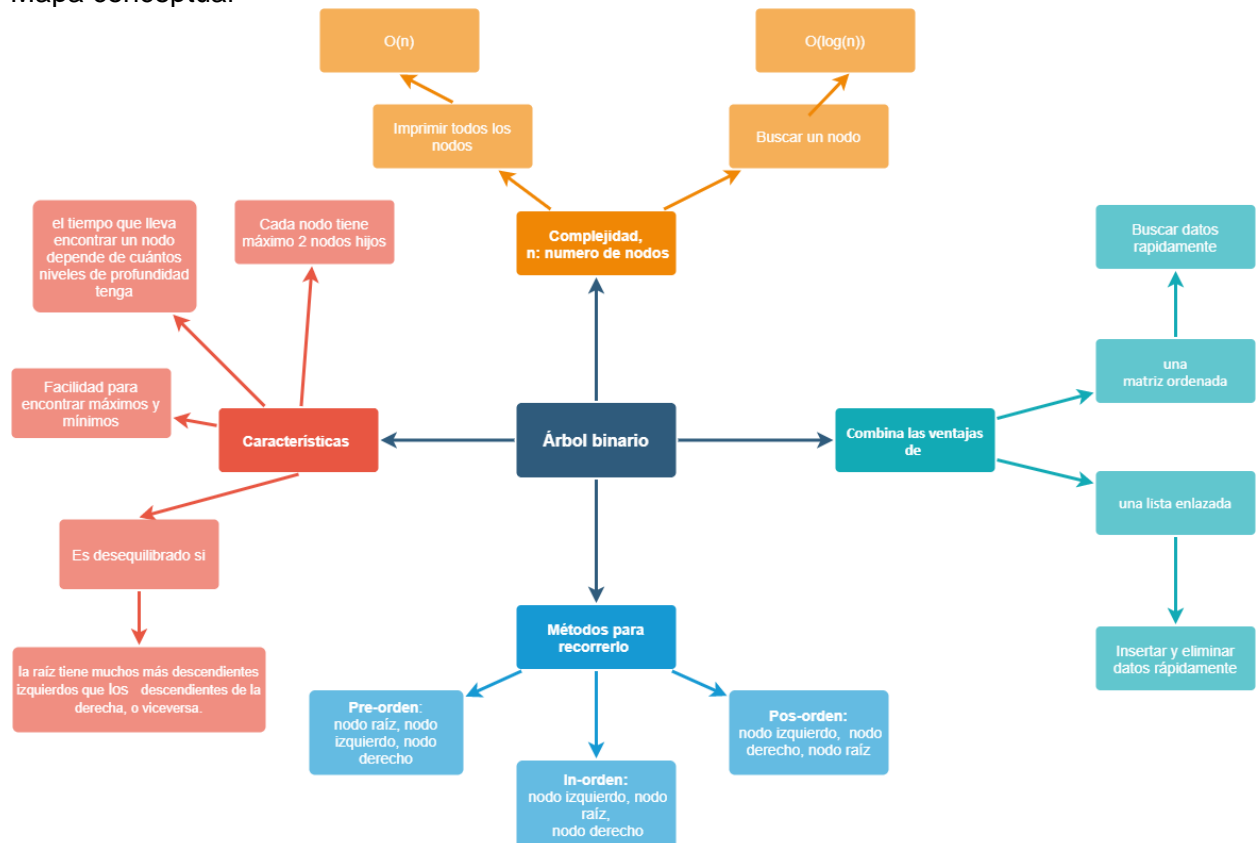
- i) A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8, I = 9, J = 10.
- a) G, D, B, A, C, E, F, I, H, J
- b) $O(\log(n))$

4.13

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

5) Lectura recomendada (opcional)

Mapa conceptual



6) Trabajo en Equipo y Progreso Gradual (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

6.1 Actas de reunión

Acta	Fecha	Integrante	Hecho
1	20/10/2020	Laura Katterine Zapata Rendón	Leer laboratorio 4
		Maria Alejandra Vélez Clavijo	Leer laboratorio 4
2	21/10/2020	Laura Katterine Zapata Rendón	4. Simulacro de Parcial 4.1 al 4.7
			4. Simulacro de Parcial 4.8 al 4.13
		Maria Alejandra Vélez Clavijo	4. Simulacro de Parcial 4.1 al 4.7
			4. Simulacro de Parcial 4.8 al 4.13
3	22/10/2020	Laura Katterine Zapata Rendón	1.1 Algoritmo abejas roboticas
			2.1 Algoritmo árbol binario de búsqueda
		Maria Alejandra Vélez Clavijo	1.1 Algoritmo abejas roboticas
			2.1 Algoritmo árbol binario de búsqueda
4	23/10/2020	Laura Katterine Zapata Rendón	Documentación Algoritmo abejas roboticas
			Documentación Algoritmo árbol binario de búsqueda
		Maria Alejandra Vélez Clavijo	Documentación Algoritmo abejas roboticas
			Documentación Algoritmo árbol binario de búsqueda
5	24/10/2020	Laura Katterine Zapata Rendón	3.1 Complejidad algoritmo 1.1
			3.4 Complejidad algoritmo 2.1
			3.5 Explicar variables complejidad
		Maria Alejandra Vélez Clavijo	3.1 Complejidad algoritmo 1.1
			3.4 Complejidad algoritmo 2.1
			3.5 Explicar variables complejidad

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

Haciendo	Por hacer
4. Simulacro de Parcial 4.1 al 4.7	4. Simulacro de Parcial 4.8 al 4.13
4. Simulacro de Parcial 4.1 al 4.7	4. Simulacro de Parcial 4.8 al 4.13
	1.1 Algoritmo abejas roboticas
	2.1 Algoritmo árbol binario de busqueda
	1.1 Algoritmo abejas roboticas
	2.1 Algoritmo árbol binario de busqueda
	3.1 Complejidad algoritmo 1.1
	Documentación Algoritmo abejas roboticas
	Documentación Algoritmo árbol binario de busqueda
	3.1 Complejidad algoritmo 1.1
	Documentación Algoritmo abejas roboticas
	Documentación Algoritmo árbol binario de busqueda
3.1 Complejidad algoritmo 1.1	3.4 Complejidad algoritmo 2.1
	3.5 Explicar variables complejidad
3.1 Complejidad algoritmo 1.1	3.4 Complejidad algoritmo 2.1
	3.5 Explicar variables complejidad
5 Lectura recomendada y realización mapa	
5 Lectura recomendada y realización mapa	

6.2 El reporte de cambios en el 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

ESTRUCTURA DE DATOS 1

Código ST0245

Historia para **ST0245-001** / laboratorios / lab04

Se compromete el 26 de octubre de 2020	
Agregar archivos mediante carga kattezapata comprometido hace 40 minutos	Verificado  80db21a <>
Agregar archivos mediante carga malejav02 comprometido hace 43 minutos	Verificado  6ad1855 <>
Se compromete el 25 de octubre de 2020	
Eliminar Main.java malejav02 comprometido hace 3 horas	Verificado  263217a <>
Eliminar Tree.java malejav02 comprometido hace 3 horas	Verificado  b49e8b3 <>
Agregar archivos mediante carga malejav02 comprometido hace 3 horas	Verificado  2f28e7f <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  abb0a31 <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  a9b7100 <>
Agregar archivos mediante carga malejav02 comprometido hace 3 horas	Verificado  2f28e7f <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  abb0a31 <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  a9b7100 <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  a20a70c <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  156f029 <>
Agregar archivos mediante carga kattezapata comprometido hace 9 horas	Verificado  883f61f <>
Ejercicio 2 malejav02 comprometido hace 9 horas	 9ec28a6 <>
Se compromete el 1 de agosto de 2017	
Agregando estructura jarcil13 comprometido on 1 Aug 2017	 a37d169 <>

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

ESTRUCTURA DE DATOS 1 Código ST0245

ESTRUCTURA DE DATOS 1
Código ST0245

Laboratorio 04
Árboles binarios y Tablas de Hash

Maria Alejandra Vélez Clavijo
Universidad Eafit
Medellín, Colombia
mavelezc1@eafit.edu.co

Nombre completo de integrante 2
Universidad Eafit
Medellín, Colombia
Correointegrante2@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

- 3.1
- 3.2
- 3.3
- 3.4
- 3.5

Mostrar cambios ☒

▼ hace 4 minutos
Versión actual

hace 5 minutos
■ Laura Katterine Zapata Rendon modificado

hace 11 minutos
■ Laura Katterine Zapata Rendon modificado

hace 19 minutos
■ Maria Alejandra Vélez Clavijo modificado

hace 1 hora
■ Laura Katterine Zapata Rendon modificado

hace 1 hora
■ Laura Katterine Zapata Rendon modificado

▼ 24 de octubre, 21:30

24 de octubre, 21:30

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

Historial de versiones

Mostrar cambios ☒

24 de octubre, 21:05

■ Maria Alejandra Vélez Clavijo modificado

24 de octubre, 21:04

■ Maria Alejandra Vélez Clavijo modificado

24 de octubre, 20:57

■ Maria Alejandra Vélez Clavijo modificado

■ Laura Katterine Zapata Rendon modificado

24 de octubre, 20:56

■ Maria Alejandra Vélez Clavijo modificado

24 de octubre, 20:42

■ Maria Alejandra Vélez Clavijo modificado

24 de octubre, 20:30

■ Maria Alejandra Vélez Clavijo modificado

24 de octubre, 20:19

■ Laura Katterine Zapata Rendon modificado

■ Maria Alejandra Vélez Clavijo modificado

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

