

Laboratorio Nro. 4 Tablas de Hash y Árboles

Alejandra Palacio Jaramillo
Universidad Eafit
Medellín, Colombia
apalacioj@eafit.edu.co

Valentina Moreno Ramírez
Universidad Eafit
Medellín, Colombia
vmorenor@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Estructura de datos implementada: árbol n-ario.

Para solucionar este problema, se diseñó un algoritmo que simula un Octree ya que, como su nombre lo indica, los 8 subnodos que tiene representan cuadrantes de las dimensiones en el espacio. Para representar este árbol, se utilizaron como estructuras base un vector dinámico y listas enlazadas ya que, en el caso de los vectores dinámicos, la complejidad para acceder a un elemento es de $O(1)$ y, en el caso de las listas enlazadas, la complejidad para insertar. Cada casilla de este vector dinámico representa un sector o cuadrante del espacio y dentro de cada una de ellas se tiene una lista enlazada que contiene las abejas que se encuentran en ese sector. Ahora, si la diagonal en cada sector es mayor que 100, se crea otro octree dentro de este nodo y se repite el proceso recursivamente. Este procedimiento se realiza con el fin de obtener como nodos terminales sectores donde la diagonal sea menor o igual a 100 para así saber cuales son los conjuntos de abejas que colisionan. Además, es importante aclarar que el código inicial para este algoritmo se tomó del código inicial dado por el profesor.

La complejidad del algoritmo es $O(n \log n)$, donde n es el número de abejas, ya que la complejidad de crear un octree es de $O(n)$ y, como se crea recursivamente pero cada vez con menos datos porque estos se dividen en 8 nodos, entonces la complejidad final sería $O(n \log n)$.

3.3 Un árbol que se muestra con una organización pre-orden, es aquel en donde se identifica primero el nodo de la raíz, para después ir indicando los valores de todo el subárbol izquierdo y cuando culmine este proceso se irá por los valores del subárbol derecho. En nuestro código, implementamos las clases, Binary Tree Node y Binary Tree para crear nuestro árbol con todos los métodos que una estructura de datos debería de tener. Para llevar a cabo los métodos recursivos, en el caso del pre-orden, se recibirá primero el valor inicial de la raíz del

ESTRUCTURA DE DATOS 1

Código ST0245

árbol, se imprimirá el valor que reciba el método para después realizar el llamado recursivo que recorrerá todos los elementos de sub-árbol izquierdo, y después se hará el llamado nuevamente, pero en este caso recorriendo los elementos del sub-árbol derecho. Cuando se realiza el método de pos-order cambia lo que cambiará será la ubicación de la función de imprimir, la cual irá al final del método en vez del inicio. Ya así tendremos nuestra manera de calcular los árboles de manera post-ordenada y pre-ordenada fácilmente.

3.5 La complejidad de todos los métodos, excepto el de building tree, es de $O(\log(n))$ ya que al estar usando árboles binarios todos los problemas que tengamos se irán dividiendo por dos. El método de building tree, tendrá una complejidad lineal $O(n)$ ya que irá recorriendo los elementos del arreglo con un ciclo for.

3.6 La variable n será el número de nodos que contiene el árbol binario dado.

4) Simulacro de Parcial

4.1

4.1.1 b

4.1.2 d

4.2 c

4.3

a. false

b. return suma == a.data

c. sumaElCamino(a.der, suma - a.data)

d. || sumaElCamino(a.izq, suma - a.data)

4.4

4.4.1 c

4.4.2 a

4.4.3 d

4.4.4 a

4.5

4.5.1 toInsert == p.data

4.5.2 toInsert > p.data

4.6

4.6.1 d

4.6.2 return 0

4.6.3 raiz.hijos.size() == 0

4.7

4.7.1 a

4.7.2 b

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

4.8 b
4.9 a
4.10 Por definir...
4.11
 4.11.1 b
 4.11.2 a
 4.11.3 b
4.12
 4.12.1 i)
 4.12.2 a
 4.12.3 a
4.13
 4.13.1 suma[e.id]
 4.13.2 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

