

Laboratorio 4

Tablas de Hash y Árboles

Juan Pablo Restrepo Escobar
Universidad Eafit
Medellín, Colombia
jrestrepo@eafit.edu.co

Juan José Sánchez Cortes
Universidad Eafit
Medellín, Colombia
jjsanchezc@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 En el punto 1.1 utilizamos las tablas de hash para poder calcular la posición de las abejas, ya que esta nos ayudaba a calcular precisamente el momento en el que dos abejas se encontraban en un mismo punto, permitiendo igualar variables de tiempo y espacio en el programa.

3.3

(2.1) En el ejercicio 2.1 nos piden un algoritmo que pueda encontrar el recorrido pre-orden lo que hacemos en el código del ejercicio 2.1 es crear la clase nodo, acá tendremos todo para crear el árbol binario. Luego, creamos diferentes métodos en los cuales podremos: insertar un valor

buscar si existe un valor en específico

imprimir los números mayores al deseado

imprimir el recorrido InOrder

imprimir el recorrido PreOrder

Orden en el algoritmo

1-imprimir el valor del nodo actual

2-moverse al nodo izquierdo (si es verdad que vuelva a empezar el método)

3-moverse al nodo derecho (si es verdad que vuelva a empezar el método)

Lo que se hace a diferencia del InOrder, es imprimir (System.out.print) el cual está al principio, ya que primero vamos a imprimir la cabeza, por lo tanto, cuando no hayan "cabezas" vamos a imprimir el nodo de la izquierda y luego el de la derecha

(2.2) El funcionamiento del 2.2 se basa en paréntesis:

1-el código empieza con un paréntesis, dentro del empieza con un número (ese número será la raíz) ejemplo de la página:

(5)

2-después del número hay otros dos paréntesis, los cuales representan la raíz de la iz y derecha. El primer paréntesis representa la izquierda. Ejemplo de la página:

(5(4)(8))

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

3-se repite el paso 2 hasta que se lleguen a lo siguiente:

(numero())()

donde los paréntesis vacíos representan que apuntan a null, por lo tanto que ya no hay rama derecha o izquierda para seguir. Ejemplo final de la página:

(5 (4 (11 (7 () ()) (2 () ())) ()) (8 (13 () ()) (4 () (1 () ()))))

Nota: puede haber casos donde solo la izquierda o derecha apuntan a null

La imagen del árbol final está en el siguiente enlace:

https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=48.

3.4 (2.1) Ya que tenemos que recorrer todo el árbol para imprimir el orden, la complejidad en el peor de los casos sería $O(n)$.

3.5 (2.1) La única variable de la cual podremos hablar es de n , donde n es el número de elementos del árbol binario ya que para poder imprimir el orden deseado, debemos de recorrer dato por dato

4) Simulacro de Parcial

4.1 B 4.1.2 D $O(1)$

4.3 a) false

b) 0

c) (a.izq, suma)

d) (a.der, suma)

4.9 A

4.13 4.13.1 raíz.hijos **4.13.2 D**

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