Laboratory practice No. 4: Hash Tables and Trees

Manuela Moreno Cordoba
Universidad Eafit
Colombia
mmorenoc2@eafit.edu.co

Yhilmar Andres Chaverra Castaño
Universidad Eafit
Colombia
yachaverrc@eafit.edu.co

3) Practice for final project defense presentation

3.1 To resolve the representation of the file system, the n-arium tree is used. This type of tree has the main characteristic that the maximum number of children per node is N, so a file system can have n number of subdirectories... With this clear, the tree search complexity would be a logarithmic function, this is because The maximum number of comparisons we would need to know if an element is in a binary search tree would be between [log2 (N 1)] and N, where N is the number of nodes.

3.3 The idea of the code is that it be able to receive an array of numbers, this will be saved in pre-order and then it will be printed in post-order, to do that, 2 classes are used, the main class Point2 and a class Node, in where it contains a left Son, a right son and the value of the data to keep. Then there are 5 methods with a main method. The main function called buildingTree () has the pre-order construction of the tree, this function calls the try method; The method called insert () has the function of inserting the left node and the right node of the tree, The method called pre-order () has the function of organizing the tree in pre-order form, The method called post-order () has as function organize the tree in post-order form, The exercise21 method has as function call the function buildingTree () and print the tree calling the post-order function (), The main method, has an example string called test where it is checked the operation of the function
3.4 T(n) = O(n)+O(log n)
3.5 int [] test = is an array of integers containing the numbers we have to order

4) Practice for midterms

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

4.2 c) 3.

4.3 a) return false;
b) return suma == a.dato ;
c)(a.izq, suma)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473

d) (a.der,suma)

4.4.1 a) T(n)=T(n−1)+C
2 a) O(n)
3 d) Wilkenson, Joaquina, Eustaquia, Florinda, Eustaquio, Jovín, Sufranio, Piolina, Wilberta, Piolín, Usnavy
4 a) Cambiar el orden de las líneas 03, 04 y 05 por 05, 04, 03

4.5 a) toInsert==null
b) toInsert<=p

4.6 1 d) 4
2 NNodo nuevo = new NNodol(raiz, suma);
3 == null

4.7 1 a) 0, 2, 1, 7, 5, 10, 13, 11, 9, 4
2 b) 2
3 d) O(n)

4.8 c) 4

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

4.10 b) no

4.11 1 b) 2, 3, 4, 0, 5, 7,
2 a) 5
3 a) Si

4.12 1 i) A = 1 B = 2 C = 3 D = 4 E = 5 F = 6 G = 7 H = 8 I = 9 J = 10
2 a) G, D, B, A, C, E, F, I, H, J
3 a) O(n)

4.13 1 raiz.id
2 a) T(n) = T(n − 1) + c, que es O(n)

PhD. Mauricio Toro Bermúdez
Professor | School of Engineering | Informatics and Systems
Email: mtorobe@eafit.edu.co | Office: Building 19 – 627
Phone: (+57) (4) 261 95 00 Ext. 9473