Laboratorio Nro. 4
Tablas de Hash y Árboles

Miguel Ángel Sarmiento Aguiar
Universidad Eafit
Medellín, Colombia
msarmie4@eafit.edu.co

Marlon Pérez Ríos
Universidad Eafit
Medellín, Colombia
mperezr@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 Para representar el sistema de archivos se utilizó un árbol de búsqueda binario, el cual tiene una complejidad de O(log(n)) para la operación de búsqueda en el caso de que el árbol se encuentre equilibrado, pero tiene una complejidad de O(n) en el caso en el que el árbol esté degenerado, el cual es el peor de los casos.

3.2 No se puede implementar más eficientemente un árbol genealógico para que la búsqueda e inserción se puedan hacer en tiempo logarítmico, ya que como se mencionó en el punto anterior, si el árbol se encuentra equilibrado las operaciones tendrían una complejidad logarítmica, pero no habría forma de relacionar los datos del árbol como se haría en la búsqueda de árboles equilibrados, la cual compara si el número buscado es mayor o menor a la raíz actual, ya que los datos son cadenas de texto relacionadas simplemente por un parentesco.

3.3 Este algoritmo es responsable de atravesar recursivamente cada nodo del árbol tanto a la izquierda como a la derecha, y cada vez que finaliza este proceso, imprime los datos almacenados en el nodo correspondiente. Primero es responsable de imprimir los datos que están a la izquierda en el nivel más bajo y subir hasta llegar a la raíz.

3.4

```
def posorden(self, node):
    if node != None:                    # C1
      self.posorden(node.left)          # C2*(n/2)
      self.posorden(node.right)         # C3*(n/2)
      print(node.data)                  # C4
```

T(n) = C1 + C2*(n/2) + C3*(n/2) + C4
T(n) = O(C1 + C2*(n/2) + C3*(n/2) + C4)
T(n) = O(2*(n/2)
T(n) = O(n)

3.5 n es ek número de nodos del árbol de búsqueda

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

4.1 **4.1.1** b
4.1.2 d

4.2 c

4.3 **a) línea 3** return false;
b) línea 5 return suma == a.data;
c) línea 7 return sumaElCamino(a.izq, suma - a.data)
d) línea 8 || sumaElCamino(a.der, suma - a.data);}

4.4 **4.4.1** c
4.4.2 a
4.4.3 d
4.4.4 a

4.5 **a) línea 4** if (p.data == toInsert)
b) línea 6 if (toInsert > p.data)

4.6 **4.6.1** d
4.6.2 línea 4 return 0;
4.6.3 línea 6 if (raiz.hijos.size() == 0)

4.7 **4.7.1** a
4.7.2 b

4.8 b

4.9 a

4.10 b

4.11 **4.11.1** b
4.11.2 a
4.11.3 a

4.12 **4.12.1** i
4.12.2 a
4.12.3 a

4.13 **4.13.1 línea 10** suma[raíz.id] = suma[e.id] + suma[raíz.id];
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