

BÚSQUEDA DE ARCHIVOS A TRAVÉS DE ESTRUCTURAS DE DATOS

Maria Paula Chaparro Muñoz
Universidad Eafit
Colombia
mpchaparrm@eafit.edu.co

Juan José Parra Díaz
Universidad Eafit
Colombia
jjparrad@eafit.edu.co

Mauricio Toro Bermúdez
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

RESUMEN

El problema a solucionar es encontrar una estructura de datos adecuada para poder buscar eficientemente un archivo determinado en una jerarquía de archivos entregada. La importancia de este problema planteado yace en el hecho de que cada día, con el avance en tecnologías y la amplia disponibilidad de información, se guardan más y más archivos en la memoria de los computadores, lo que conlleva a que sea cada vez más complicado encontrar algún archivo específico.

Palabras clave

Algoritmo de búsqueda, Estructuras de datos, Complejidad, Manejo de datos, Accesibilidad, Software.

Palabras clave de la clasificación ACM

•Theory of computation ~ Data structures and algorithms for data management •Theory of computation ~ Theory and algorithms for application domains •Theory of computation ~ Database theory

•Information systems ~ Data management systems
•Information systems ~ Data access methods
•Information systems •Information systems ~ Data structures

1. INTRODUCCIÓN

En el tiempo en el que estamos, la era de la información, nos vemos bombardeados por miles de archivos, datos, números y letras, llenando nuestros computadores. Con la llegada del internet cada día hay algo nuevo, nuevos bits de información y nuestra necesidad de almacenar archivos es cada vez mayor. Necesitamos más memoria, más espacio, lo que conlleva a que tengamos que recurrir a estructuras de datos que nos faciliten la representación de directorios y archivos, y que además de esto, nos permita hacer búsquedas, de manera que podamos acceder eficientemente a todos nuestros documentos.

2. PROBLEMA

El problema consiste en realizar una estructura de datos que pueda representar de manera eficiente los archivos y directorios dentro de un computador y que permita hacer búsquedas dentro de éste velozmente.

3. TRABAJOS RELACIONADOS

Problemas similares a este problema de organización de archivos y directorios a través de estructura de datos son muy comunes, puesto que todos los computadores en el

mundo poseen esta función. Por este motivo, existen soluciones en manera de estructuras de datos que pueden ser potencialmente las respuestas a este problema.

3.1 Árbol AVL

Esta estructura de datos como solución al problema planteado consiste en una organización de datos en manera de árbol, el cual tiene un nodo raíz, y sus respectivos nodos hijos. La característica principal de esta estructura de datos como solución, es el hecho de que es un árbol equilibrado, es decir, que la diferencia de alturas entre los nodos extremos es igual o menor a 1 (mínima). Esto permite que esta solución sea una rápida y efectiva a la hora de encontrar la información requerida, puesto que la distancia que tiene el nodo raíz con respecto a todos sus nodos es la más corta posible¹.

3.2 Árbol B

Reciben su nombre de R. Bayer, quien en 1970 propuso un nuevo tipo de árboles, en los que todas las páginas, excepto una, contienen entre n y $2n$ nodos, siendo n una constante dada. Estos árboles no son binarios a diferencia de los AVL o los de búsqueda, constituyen una categoría muy importante de estructuras de datos, que permiten una implementación eficiente de conjuntos y diccionarios, para operaciones de consulta y acceso secuencial.

En estos árboles los nodos se agrupan en páginas y las páginas por lo tanto pueden ser definidas como un conjunto de nodos²

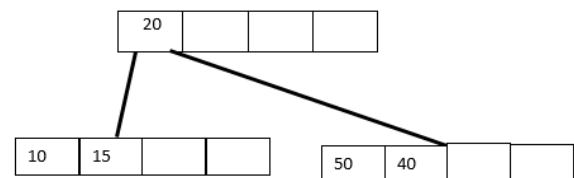


Gráfico 1:

Ejemplo gráfico de árbol, aquí insertamos como valor unos números enteros

3.3 Árbol Rojo Negro

El árbol rojo negro es una estructura de datos principalmente de búsqueda, la cual está fundamentada en una serie de reglas de color, lo que permite desplazarse de

¹ (Rodríguez, 2011)

² (Villanueva, s.f.)

una marea veloz entre los diferentes nodos, identificando el color de cada uno. Esta forma de búsqueda permite que se logre encontrar una aproximación a un equilibrio en la estructura. De igual manera, gracias a la eficiencia de esta solución, el camino más largo hacia la raíz es igual a dos veces el camino más corto, haciendo esta estructura una muy veloz.³

3.4 Tablas de Hash

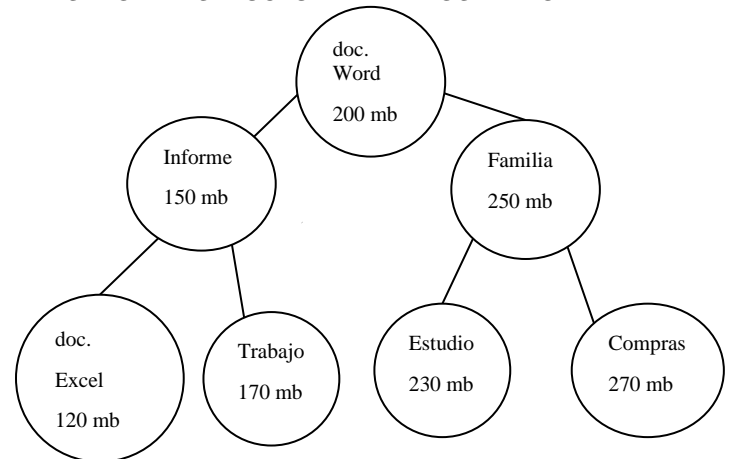
Las tablas hash son estructuras de datos que se utilizan para almacenar un número elevado de datos sobre los que se necesitan operaciones de búsqueda e inserción muy eficientes.

Una tabla de hash se puede ver como un conjunto de entradas. Cada una de estas entradas tiene asociada una clave única y por lo tanto, diferentes entradas de una misma tabla tendrán diferentes claves. Otra definición un poco más compleja se puede expresar diciendo que las Tablas de hash son técnicamente contenedores asociativos (tipo Diccionario) que permiten un almacenamiento y posterior recuperación de elementos (valores) a partir de otros objetos, llamados claves⁴.

Pablo	0
Daniela	1
Paula	2
Laura	3
Ana	4

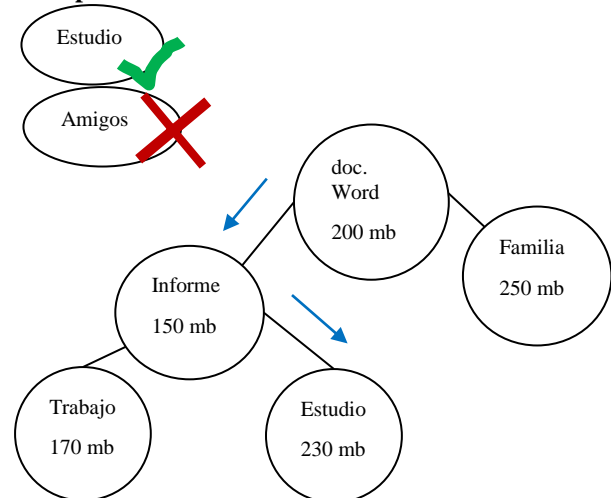
Gráfico 2: Ejemplo de una tabla de Hash

4. DISEÑO DE ESTRUCTURA DE DATOS: ÁRBOL AVL

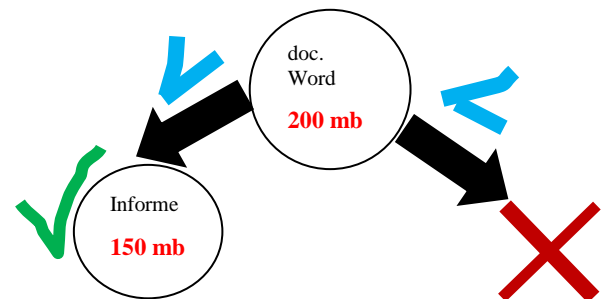


Gráfica 3: Organiza en un árbol AVL unos documentos de muestra de acuerdo con su tamaño

4.1 Operaciones del árbol AVL



Gráfica 4: Operación de recorrer el árbol junto con comparación usado para búsqueda del archivo



Gráfica 5: Operación de adición de un elemento al árbol AVL comparándolo con el nodo padre y ubicándolo respectivamente a la derecha o izquierda

³ (Universidad de Valladolid, 2011)

⁴ (Universidad Carlos III de Madrid, s.f.)

4.2 Criterios de diseño de la estructura de datos

Se decidió que el diseño de la estructura de datos que se implementa para este problema sea a partir de un árbol AVL. Este tipo de estructura es un árbol binario que

permite una búsqueda muy eficiente con una complejidad $O(\log_n)$, es decir, tiene una complejidad muy baja con respecto a otras estructuras de datos. Esto, debido a que es un árbol que tiene como principal característica, la poca diferencia de altura entre los nodos extremos, lo que conlleva a que el recorrido en este sea considerablemente veloz. Se eligió esta estructura de datos pues tiene muchas ventajas con respecto a otras: tiene una complejidad muy baja en todas las operaciones comparado con una lista enlazada y a la hora de buscar permite descartar numerosos elementos con solo una comparación.

Se eligió hacer la estructura a partir de un árbol AVL y no con una tabla de hash debido a que esta última es ineficiente cuando se produce una colisión, caso que no ocurre en los árboles AVL. De igual manera, se podría considerar el uso de un árbol rojo-negro en vez de un árbol AVL, puesto que estos son más veloces a la hora de insertar y borrar datos, pero no es el caso óptimo para el problema planteado, debido a que la idea de la estructura es permitir al usuario encontrar archivos de manera veloz, no crearlos o destruirlos, situación en la que un árbol AVL es superior a un árbol rojo-negro. Es por estas razones que se consideró que el uso de un árbol AVL era la mejor estructura de datos posible para la solución del problema planteado. Se espera que con el uso de esta se logre la solución más eficiente posible a la hora de buscar la información que se desea encontrar.

REFERENCIAS

1. Rodríguez, C. (2011). Estructuras de Datos y Algoritmos. Obtenido de Departamento de Informática Universidad de Valladolid:
<https://www.infor.uva.es/~cvaca/asigs/doceda/tema4.pdf>
2. Universidad Carlos III de Madrid. (s.f.). Arquitectura de Sistemas. Obtenido de Departamento de Ingeniería Telemática:
http://www.it.uc3m.es/abel/as/MMC/M2/HashTable_es.html
3. Universidad de Valladolid. (2011). Definición de un árbol Rojinegro. Obtenido de Departamento de Informática Universidad de Valladolid:
<https://www.infor.uva.es/~cvaca/asigs/doceda/rojonegro.pdf>
4. Villanueva, S. S. (s.f.). DataStructures Tool. Obtenido de Human Communication and Interaction Research Group:
<http://www.hci.uniovi.es/Products/DSTool/b/b-queSon.html>