

# Informe Proyecto

Calzada Matínez Jonathan Omar, García Lazcano Carlos David, Hernández Jiménez Juan Carlos

May 11, 2019

## Estructura de Datos y Algoritmos II

Grupo: 5

Semestre: 2019-2

Profesor: M.C. Tista García Edgar

### Integrantes:

TEMA: Algoritmos Árboles binarios de búsqueda equilibrados (AVL) y arboles B más.

TEMA: Algoritmos Árboles binarios de búsqueda equilibrados (AVL) y arboles B más.

Introducción: Un Tipo de dato abstracto (en adelante TDA) es un conjunto de datos u objetos al cual se le asocian operaciones. El TDA provee de una interfaz con la cual es posible realizar las operaciones permitidas, abstrayéndose de la manera en como estén implementadas dichas operaciones. Esto quiere decir que un mismo TDA puede ser implementado utilizando distintas estructuras de datos y proveer la misma funcionalidad.

El paradigma de orientación a objetos permite el encapsulamiento de los datos y las operaciones mediante la definición de clases e interfaces, lo cual permite ocultar la manera en cómo ha sido implementado el TDA y solo permite el acceso a los datos a través de las operaciones provistas por la interfaz.

En este capítulo se estudiarán TDA básicos como lo son las listas, pilas y colas, y se mostrarán algunos usos prácticos de estos TDA.

Planteamiento del problema El estudio de los arboles b y b+ es un tema de importancia, ya que podría facilitar mucho la forma en la que se resuelven problemas computacionales en relación a las bases de datos para poder ejecutar las queries más rápido. El sistema responde más rápido porque no tiene que hacer validaciones por cada elemento, sino más bien por bloques, si el criterio especificado entra en el rango de este para sustraerlo. Estudiar y entender los arboles Binarios de búsqueda equilibrados y los arboles B+.

Hipótesis: Los árboles B tienen ventajas sustanciales sobre otras implementaciones cuando el tiempo de acceso a los nodos excede al tiempo de acceso entre nodos. Este caso se da usualmente cuando los nodos se encuentran en dispositivos de almacenamiento secundario como los discos rígidos. Objetivo: Que el alumno implemente árboles las estructuras de datos de tipo árbol y que desarrolle sus habilidades la programación orientada a objetos a través de la aplicación del concepto de árboles como estructuras de datos no lineales

¿Qué es un Árbol Binario? ÁRBOLES. ÁRBOLES BINARIOS. Hasta ahora nos hemos dedicado a estudiar TAD es que de una u otra forma eran de naturaleza lineal, o unidimensional. En los tipos abstractos de datos lineales existen exactamente un elemento previo y otro siguiente (excepto para el primero y el último, si los hay); en las estructuras no lineales, como conjuntos o árboles, este tipo de secuencialidad no existe, aunque en los árboles existe una estructura jerárquica, de manera que un elemento tiene un solo predecesor, pero varios sucesores. Una exploración algo amplia en el campo de la ciencia de la computación nos lleva a situaciones en que las representaciones lineales son inadecuadas, tanto en sentido conceptual como práctico. Un paso importante lo representan los árboles binarios, y el siguiente vendrá dado con el estudio de la noción general de árbol. En capítulos posteriores, lo extenderemos hasta llegar a los grafos. Un árbol impone una estructura jerárquica sobre una colección de objetos. Ejemplos claros de utilización de árboles se presentan tanto dentro como fuera del área de computación (índices de libros, árboles genealógicos, etc.); en Informática constituyen una de las estructuras más utilizadas, con aplicaciones que van desde los árboles sintácticos utilizados para la representación y/o interpretación de términos de un lenguaje o expresiones aritméticas, pasando por los arboles de activación de procedimientos recursivos, hasta la representación de datos que se desea mantener ordenados con un tiempo de acceso relativamente bajo. En general, se usarán árboles siempre que se quiera representar información jerarquizada, cuando esta converja en un solo punto.

Se define un árbol binario como un conjunto finito de elementos (nodos) que bien esta vacío o esta formado por una raíz con dos arboles binarios disjuntos, es decir, dos descendientes directos llamados subarbol izquierdo y subarbol derecho.

Los árboles binarios (también llamados de grado 2 )tienen una especial importancia. Las aplicaciones de los arboles binarios son muy variadas ya que se les puede utilizar para representar una estructura en la cual

disco lógicamente el tiempo resultante de ejecución de nuestra búsqueda se ve fuertemente recortado. Por consiguiente, si tratamos de construir una estructura de datos sobre disco es fundamental tener en cuenta que uno de los factores determinantes en el tiempo de ejecución es el número total de accesos, de forma que aunque dicho número mero pueda ser acotado por un orden de eficiencia es muy importante tener en cuenta el número real ya que el tiempo para realizar un acceso es suficientemente alto como para que dos algoritmos pese a tener un mismo orden, puedan tener en un caso un tiempo real de ejecución aceptable y en otro inadmisiblemente. De esta forma, si construimos un árbol binario de búsqueda equilibrado en disco, los accesos a disco serán para cargar en memoria uno de los nodos, es decir, para poder llevar a memoria una cantidad de información suficiente como para poder decidir entre dos ramas. Los árboles de múltiples ramas tienen una altura menor que los árboles binarios pues pueden contener más de dos hijos por nodo, además de que puede hacerse corresponder los nodos con las páginas en disco de forma que al realizar un único acceso se leen un número alto de datos que permiten elegir un camino de búsqueda no entre dos ramas, sino en un número considerablemente mayor de ellas. Además, este tipo de árboles hace más fácil y menos costoso conseguir equilibrar el árbol.

En resumen, los árboles con múltiples hijos hacen que el mantenimiento de índices en memoria externa sea mucho más eficiente y es justamente éste el motivo por el que este tipo de árboles han sido los que tradicionalmente se han usado para el mantenimiento de índices en sistemas de bases de datos. Lógicamente, aunque este tipo de estructuras sean más idóneas para mantener grandes cantidades de datos en almacenamiento externo es posible construirlas de igual forma en memoria principal, y por consiguiente pueden ser mantenidas en memoria (mediante el uso de punteros, por ejemplo) al igual que las que hemos estudiado hasta ahora.

Metodología

Para poder realizar el árbol Binario de búsqueda equilibrado. Se comenzó a analizar el algoritmo realizado para la práctica nueve de laboratorio en donde de ahí partimos para realizar el árbol binario de búsqueda balanceado. Para el algoritmo de el árbol B más se analizó de igual manera el árbol B más proporcionado por el profesor y de ahí se partió para poder implementar el árbol B+.

Conclusiones: Referencia: