Análisis de busquedas en estructuras de datos

Luis Carlos Quesada R. B65580

***Resumen*—Mediante el conteo de las búsquedas de datos en las estructuras *Árbol de búsqueda binaria* y *Lista enlazada* en un tiempo determinado, se obtienen resultados que permiten de manera acertada, averiguar que tan apropiada es una estructura para determinado caso de datos.**

Con los resultados de la investigación, se conoce que los arboles de búsqueda binaria son apropiados para reducir la cantidad de consultas dentro de la estructura, sin embargo para casos de inserción ordenada no es apropiado utilizar este modelo, también se conoce que una lista enlazada posee tiempos de inserción y borrado sencillos, pero tiempos de búsqueda más complejos que el árbol, sin embargo utilizando menor espacio que un árbol de búsqueda en el peor caso.

I. Introducción

Las estructuras de datos nos permiten organizar, representar y manejar conjuntos de datos. Con las estructuras es posible facilitar el uso y manipulación de conjuntos de datos de gran tamaño en forma sencilla. A pesar de que cada una de estas estructuras cumple con un objetivo común, se implementan de maneras bastante diversas, y así mismo se diversifican sus comportamientos. En la presente investigación se analizó la cantidad de búsquedas de elementos dentro de las estructuras de datos: Árbol de búsqueda binaria y Lista enlazada, en un periodo de tiempo dado.

II. Metodología

La prueba se realizó en una PC con sistema operativo Linux de 64 bits, ocho Giga-bytes de memoria RAM y procesador Intel i5 de cuarta generación, para calcular el tiempo de la prueba se utilizó la biblioteca *CTime*.

Para llevar a cabo el análisis de búsquedas mencionado en la sección I se llevó a cabo una prueba de la siguiente manera:

Inicialmente se implementan las estructuras Lista y Árbol, según la implementación encontrada en el libro *Introduction to algorythms third edition*[1]. A continuación en un método de prueba se crean dos instancias de cada estructura, a la primer instancia tanto del árbol como la lista se les insertan un millón de nodos ordenados, cada uno con un entero dentro del rango [0, 999.999], inmediatamente con las 2 instancias restantes se procede a insertar un millón de elementos nuevamente, sin embargo estos elementos son escogidos aleatoriamente en el rango [0, 2.000.000]. Antes de obtener los resultados se realiza un paso final, se evalúa la cantidad de búsquedas tanto fallidas como exitosas en cada instancia dentro de un periodo de diez segundos, para finalmente imprimir el total de resultados de la prueba.

III. Resultados

Durante la realización de la prueba se obtuvo como resultado que la cantidad de búsquedas realizadas en una lista enlazada en el tiempo de 10 segundos es similar tanto en el caso de elementos ordenados como en el caso de elementos desordenados, tal como se observa en la tabla I, por este motivo es razonable pensar que el tiempo de ejecución será el mismo sin importar el orden de los elementos.

Cuadro I
Busquedas realizadas en 10 segundos - Lista Enlazada

Inserción	Exitosas	Fallidas	Total
Ordenada	1002	1609	2611
Aleatoria	1338	1348	2686

Cuadro II
Busquedas realizadas en 10 segundos - Arbol de busqueda binario

Inserción	Exitosas	Fallidas	Total
Ordenada	176964	272266	449230
Aleatoria	1098	1099	2197

Al realizar las búsquedas en el árbol de búsqueda binario se presentaron algunas complicaciones, la inserción de nodos ordenados en un árbol es increíblemente lenta, además de que es posible que se presenten *overflows* de memoria en implementaciones recursivas. La cantidad de búsquedas en un periodo de 10 segundos para el caso del árbol aleatorio son considerablemente mayores [1] a la cantidad de búsquedas en el árbol ordenado, tal como se nota en la tabla II, este ultimo comparte un resultado similar a las listas ordenadas, de esta forma confirmando que el tiempo de búsqueda en esta estructura y caso particular, el tiempo de inserción es de $O(n)$.

IV. Conclusiones

Con los resultados de la investigación se puede concluir que el árbol de búsqueda binario puede reducir la complejidad de una búsqueda en el caso promedio y así dar muy buenos resultados, sin embargo, la inserción de elementos ordenados a este modelo de datos provoca que su tiempo de ejecución escale rápidamente de forma lineal. Se concluye que la lista enlazada es rápida en la inserción y borrado de sus elementos, sin embargo el tiempo de búsqueda deja mucho que desear, también se concluye que el tiempo de búsqueda en una lista enlazada y en un árbol de ordenado es similar. Se encuentra también que en el caso de árbol ordenado, se da un desperdicio grande en memoria, causando incluso problemas de *Overflow* en la memoria, principalmente en el destructor de la estructura, ya que es realmente complejo realizar este proceso de manera

[1] 99.5 % de las búsquedas totales en la estructura de árbol, se dan en el caso de árbol con inserción de nodos aleatorios

iterativa sin consumir la misma cantidad de memoria en una estructura auxiliar como sería una pila.

V. Bibliografía

Referencias

[1] Thomas H. Cormen. *Introduction to algorythms third edition* Masachussets Institute of Techonology, 2009.

Luis Carlos Quesada Estudiante de Bachillerato en Computación e informática de la Universidad de Costa Rica, interesado principalmente en el software libre, inteligencia artificial, robótica y sistemas descentralizados.