

Tarea 2

Daniel Monge Arce - B85017

I. INTRODUCCIÓN

EN este trabajo se implementan las estructuras de datos que corresponden al segundo cuarto del curso, y se comprueba sus diferencias. Para esta primera entrega se realizaron la Lista enlazada con nodo centinela, el Árbol de búsqueda binario, el Árbol Rojo-Negro y las Tablas de Dispersión. Los códigos están escritos en lenguaje de programación C++, y ejecutados con un main que no viene incluido con los códigos, pero inserta en una lista vacía de 1 000 000 de nodos con enteros aleatorios y realiza una búsqueda registrando la cantidad de búsquedas que se realizaron en 10 segundos para Listas enlazadas. Por otro lado, para Árboles de búsqueda binarios se insertaron 1 000 000 de nodos con un árbol de búsqueda binario con la misma cantidad de llaves y se compara cual de las 2 estructuras de datos es sustancialmente mejor que la otra (más del doble de búsquedas).

II. METODOLOGÍA

Para realizar el análisis, se realizaron las pruebas de la siguiente forma: Se hace una inserción de 1 000 000 de elementos ya ordenados en la estructura de datos que utilizamos, ya sea el árbol o la lista, una vez insertados, se realizan todas las búsquedas posibles en un lapso de 10 segundos con un cronómetro encontrado en la biblioteca CTime. Después de haber tomado el tiempo de la búsqueda, la estructura elimina todos sus contenidos y se vuelve a llenar con 1 000 000 de elementos, pero esta vez, se utilizan elementos aleatorios de rango [0, 1999999], y se vuelven a hacer cuantas búsquedas sean posibles en el mismo lapso de 10 segundos. De esta forma obtenemos la cantidad de búsquedas realizadas con elementos ordenados y aleatorios. Las pruebas fueron realizadas en Windows 10, con mingw32 para la primera parte, para la segunda parte se utilizó Ubuntu con g++, además cabe denotar que los resultados se dieron en una computadora de escritorio con un procesador: Intel(R) Core(TM)i5-6500 CPU @ 3.20GHz, 3201 Mhz, 4 Core(s), 4 LogicalProcessor(s), por lo que los tiempos pueden variar según la máquina que se utilice.

III. RESULTADOS

Los tiempos de ejecución de las X corridas de los algoritmos se muestran en el cuadro I.

Los resultados de la lista enlazada muestran cantidades de búsquedas parecidas ya sea con inserción ordenada o aleatoria, pero las aleatorias lograron hacer más búsquedas exitosas y menos fallidas que las ordenadas, y también se hicieron más pruebas, 75 pruebas más para la aleatoria para ser exactos. Pero no se notan una diferencia muy marcada entre las 2 inserciones, es decir no sobre pasan por el doble o más. Por otro lado, usando el árbol de búsqueda binaria si ve

Cuadro I
TABLA DE RESULTADOS DE LAS ESTRUCTURAS.

Estructura	Insercion	Resultados 1 millon de nodos en 10s		
		Exitosas	Fallidas	Total
Lista Enlazada	Ordenada	1668	2275	3943
	Aleatoria	2004	2014	4018
Arbol de busqueda Binario	Ordenada	293760	389062	682822
	Aleatoria	1764	1765	3529
Arbol Rojo-Negro	Ordenada	353339	352707	706046
	Aleatoria	273880	421579	695459
Tabla de Dispersion	Ordenada	361338	361272	722610
	Aleatoria	284830	439651	724481

una diferencia muy notable entre los tipos de inserciones, se realizaron casi 200 veces más búsquedas ordenadas que aleatorias, es decir, mucho más eficiente en cuanto a búsquedas totales en el lapso de 10 segundos. También vemos que la cantidad de búsquedas exitosas y fallidas son casi las mismas para las aleatorias, mientras que las ordenadas se ve un incremento importante en la cantidad de fallidas, es decir una diferencia de 95302 búsquedas fallidas más que exitosas. Lo que corresponde al Árbol Rojo-Negro se puede notar que hizo una cantidad de búsquedas parecidas al árbol binario ordenado, sin embargo, con datos aleatorios se realizaron muchas más búsquedas que el binario. La cantidad de búsquedas con inserción ordenada y aleatoria también son casi idénticas, por una diferencia de menos de 11 mil búsquedas. Por otro lado, vemos que las búsquedas fallidas son bastante notables, casi la misma cantidad de exitosas en inserción ordenada y visiblemente mayor en las aleatorias. Pero siempre más rápido que búsqueda binario. Y para tabla de dispersión, fue la estructura que más búsquedas realizó de las 4 estructuras que usamos, pero presenta datos casi idénticos al Árbol Rojo-Negro, pude notar que siempre existieron más búsquedas fallidas que exitosas en los dos casos de inserción, no obstante los casos exitosos fueron mayores que el Árbol Rojo-Negro también en ambas inserciones.

IV. CONCLUSIONES

A partir de los resultados se puede concluir que la estructura de tablas de dispersión con elementos ordenados hizo más búsquedas exitosas que las 4 estructuras, sin embargo, no por una diferencia muy mayor a las demás. También es la estructura que más búsquedas totales hizo en los 10 segundos, pero también la que más búsquedas fallidas obtuvo. Se puede ver que siempre hay más búsquedas fallidas que exitosas cuando la inserción es aleatoria, en cualquiera de las 4 estructuras, mientras que para inserción ordenada, el Árbol Rojo-Negro y la Tabla de Dispersión obtuvieron más búsquedas

exitosas con inserción ordenada. Los Árboles Rojo-Negros y la Tabla de Dispersión son muy eficientes, sin embargo son muy complicadas de programar, tiene un nivel de complejidad mucho mayor al Árbol de búsqueda binario y Lista Enlazada.

Los codigos estan incluidos en la carpeta donde se encuentra este informe.

REFERENCIAS

Sitios utilizados para entender mejor los algoritmos:

<https://www.cs.usfca.edu/galles/visualization/StackLL.html>

<https://www.cs.usfca.edu/galles/visualization/BST.html>

<https://www.cs.usfca.edu/galles/visualization/RedBlack.html>

<https://www.cs.usfca.edu/galles/visualization/OpenHash.html>

<https://www.cs.usfca.edu/galles/visualization/ClosedHash.html>