

# Tarea 2: Análisis de Algoritmos

Rubén Hernández A01024669  
Carla Pérez Gavilán A01023033  
Christian Alberto Dalma A01423166

Marzo 22, 2020

## 1 Resumen

En el siguiente reporte se desarrollan implementaciones de un árbol B y un árbol AVL, con el objetivo de realizar una comparación en los tiempos de inserción, borrado y búsqueda entre ambos. De forma que se pueda evaluar las ventajas y desventajas de la implementación de una estructura de datos en disco (externa) o en memoria (interna). Así como identificar los factores que permiten la mejora de su eficiencia. Por ejemplo, pequeñas alteraciones en su implementación.

## 2 Introducción

### Árbol B

Un árbol B es aquel que agrupo de forma ordenada a sus nodos en estrcuturas de tamaño  $N$ , que se les puede denominar páginas.

Todos los árboles B deben de cumplir con las siguientes características:

- Siendo  $n$ , la cantidad mínimo de nodos por página;  $2n$  será la cantidad máxima de nodos al que también se le denominará la letra  $N$ .
- Todos las páginas deben cumplir con la condición de tener al menos  $2n$  nodos, a excepción de la raíz que podrá tener como mínimo 2 nodo.
- Todos los nodos deben cumplir con un orden ascendente de izquierda a derecha. Por lo tanto todo nodo a la derecha de otro, es menor a este.
- Cada nodo apunta hacia otros dos páginas, la página izquierda tendrá nodos menores a este y la derecha mayores.
- Se le denomina nodo hoja a aquel cuyos apuntadores son todos nulos.

NOTA: Para que se pueda impementar una estructura como esta en disco se debe hacer uso de archivos binarios, de forma que en lugar de apuntadores

seamos capaces de acceder a los hijos de un nodo padre por medio de índices que indiquen la posición en el archivo del nodo hijo.

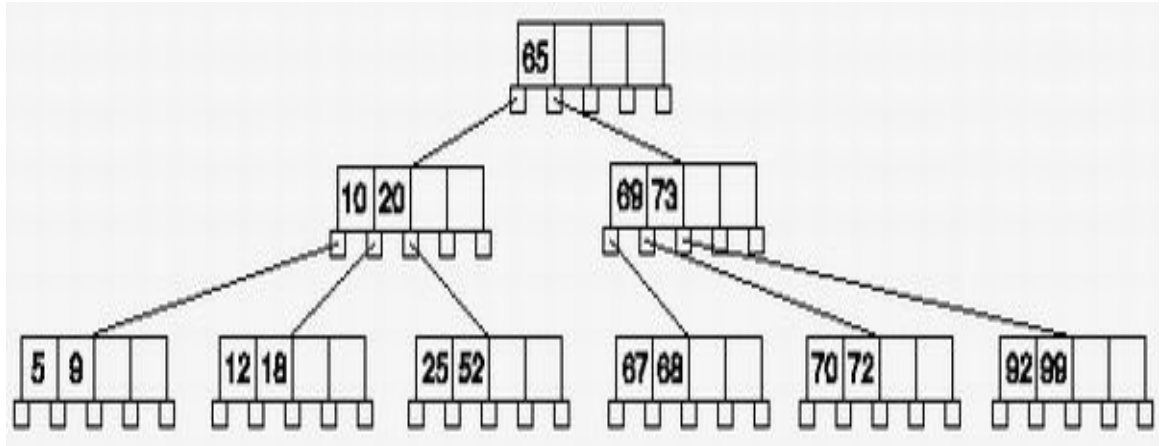


Figure 1: Ejemplo de árbol B donde  $n=2$ , y por lo tanto  $N=4$

## Árbol AVL

Un árbol AVL consiste en un árbol binario de búsqueda, que debe cumplir con la condición de siempre estar balanceado. Es por ello que debe cumplir con las siguientes características:

- Por ser un ABB cumple con la condición de que todos sus subárboles izquierdos deben ser menores a los del subárbol derecho, es por ello que los nodos siempre permanecen en orden ascendente de izquierda a derecha.
- El mantenerse balanceado consiste en que el factor de balanceo entre el árbol derecho y el árbol izquierdo (calculado por la diferencia entre la altura de uno menos la altura del otro) debe ser a lo sumo 1.
- Después de una inserción se debe retornar al estado de balanceo.

## 3 Desarrollo

Al correr el código en una Raspberry Pi 3, con procesador de 64 bits y cuatro núcleos se obtienen los siguientes tiempos de ejecución:

### Mediciones para Creación de Árboles

A continuación se presenta el tiempo de ejecución de cada una de las tres operaciones en milisegundos y nanosegundos:

Table 1: Comparación de Inserción

N	AVL ( $10^{-3}s$ )	Árbol B ( $10^{-9}s$ )
10	0.03	510334
100	0.52	40900197
1,000	3.44	0
10,000	36.60	0
100,000	292.02	0
1,000,000	500.47	0

### Mediciones Arbol AVL

A continuación se presenta el tiempo de ejecución de cada una de las operaciones en milisegundos:

Table 2: 10 Elementos

Elementos	Tiempo de Búsqueda	Tiempo de Eliminación
0	0.005	0.0183
1	0.019	0.0029
2	0.014	0.0035
3	0.023	0.0032
4	0.018	0.0031
5	0.018	0.0025
6	0.016	0.0034
7	0.019	0.003
8	0.021	0.0033
9	0.02	0.0028
10	0.016	0.003
Total	0.189	0.049

Table 3: 100 Elementos		
Elementos	Tiempo de Búsqueda	Tiempo de Eliminación
0	0.021	0.0094
1	0.021	0.0082
2	0.024	0.007
3	0.023	0.0062
4	0.022	0.0041
5	0.018	0.0036
6	0.0179	0.0031
7	0.025	0.0035
8	0.03	0.0034
9	0.04	0.0034
10	0.019	0.0027
Total	0.2609	0.0546

Table 4: 1,000 Elementos		
Elementos	Tiempo de Búsqueda	Tiempo de Eliminación
0	0.024	0.0195
1	0.023	0.019
2	0.025	0.0192
3	0.02	0.0191
4	0.019	0.0195
5	0.018	0.0194
6	0.0021	0.0199
7	0.022	0.0198
8	0.021	0.0199
9	0.028	0.02
10	0.03	0.02
Total	0.2321	0.2153

Table 5: 10,000 Elementos

<b>Elementos</b>	<b>Tiempo de Búsqueda</b>	<b>Tiempo de Eliminación</b>
8980	0.024	0.286
9005	0.028	0.042
6223	0.026	0.046
707	0.031	0.049
69	0.036	0.039
71	0.028	0.043
72	0.034	0.04
517	0.033	0.051
1214	0.03	0.084
813	0.024	0.042
737	0.026	0.043
Total	0.32	0.765

Table 6: 100,000 Elementos

<b>Elementos</b>	<b>Tiempo de Búsqueda</b>	<b>Tiempo de Eliminación</b>
8980	0.039	0.023
9005	0.037	0.043
6223	0.036	0.046
707	0.03	0.05
69	0.027	0.061
71	0.037	0.034
72	0.034	0.033
517	0.032	0.066
1214	0.038	0.069
813	0.15	0.066
737	0.159	0.042
Total	0.619	0.533

Table 7: 1,000,000 Elementos		
Elementos	Tiempo de Búsqueda	Tiempo de Eliminación
8980	0.038	0.189
9005	0.037	0.037
6223	0.041	0.051
707	0.035	0.218
69	0.032	0.041
71	0.04	0.041
72	0.036	0.036
517	0.037	0.032
1214	0.031	0.046
813	0.034	0.043
737	0.033	0.037
Total	0.394	0.771

## Mediciones

A continuación se presenta el tiempo de ejecución de cada una de las tres operaciones en nanosegundos:

Table 8: Comparación de Inserción		
N	AVL ( $10^{-9}s$ )	Árbol B ( $10^{-9}s$ )
10	indef	510334
100	indef	40900197
1,000	indef	0
10,000	indef	0
100,000	indef	0
1,000,000	indef	0

Table 9: Comparación de Búsqueda		
N	AVL	Árbol B
10	10	80580
100	100	248104
1,000	23.113231	c
10,000	23.113231	d
100,000	23.113231	e
1,000,000	23.113231	f

Table 10: Comparación de Eliminación

<b>N</b>	<b>AVL</b>	<b>Árbol B</b>
10	10	380698
100	100	268161
1,000	23.113231	c
10,000	23.113231	d
100,000	23.113231	e
1,000,000	23.113231	f

## Gráficas

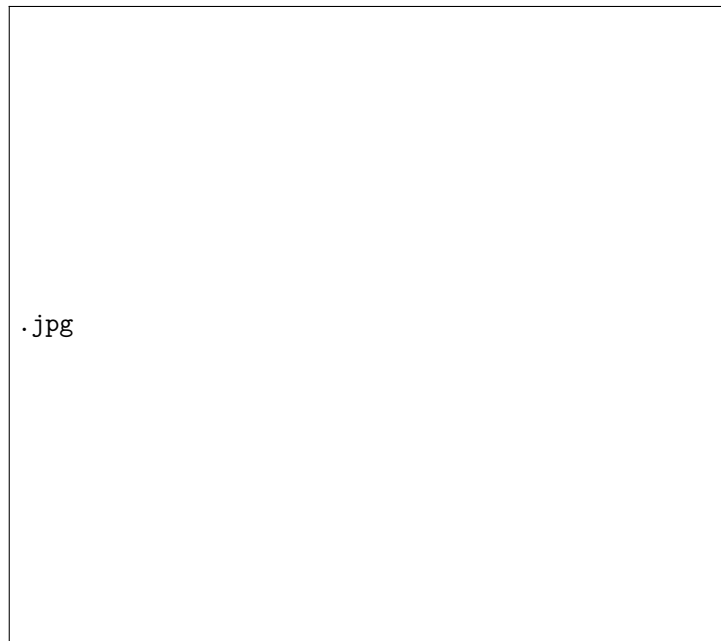


Figure 2: Gráfica comparativa de inserción

Figure 3: Gráfica comparativa de búsqueda

Figure 4: Gráfica comparativa de eliminación

## 4 Conclusión

## 5 Referencias Bibliográficas

Aquellas utilizadas para el manejo de archivos de acceso aleatorio:

- <http://c.conclase.net/ficheros/?cap=004>