



Ciencias de la
Computación
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Tarea 3 - Diccionarios

CC4102 - Diseño y Análisis de Algoritmos

Alumnos : Sebastián González
 : Patricio Isbej
Profesor : Gonzalo Navarro
Auxiliar : Jorge Bahamonde
Fecha : 16 de diciembre de 2015

Resumen

En este trabajo se analizó 4 estructuras distintas para diccionarios, implementandolas, ejecutandolas para distintos casos y comparando para uno su rendimiento ante inserciones, consultas y eliminaciones.

1. Introducción

Para este trabajo se compararon los siguientes tipos de diccionarios

- Árbol de Búsqueda Binaria convencional
- Árbol AVL

Árbol de Búsqueda Binaria convencional consiste en el árbol donde los elementos son agregados y eliminados a medida que se pide, sin realizar ningún tipo de modificación u optimización cuando se hacen las operaciones.

Árbol AVL revisa al momento de insertar o eliminar elementos si los hijos de cada nodo involucrado difieren en sus tamaños por más de 1, en caso de ocurrir, realiza un rebalanceo de manera que se cumpla la condición

Para cada estructura se realizaron los siguientes tests

- 2^4 a 2^{14} inserciones
- 2^4 a 2^{15} búsquedas para las estructuras de mayor tamaño
- 2^4 a 2^{15} borrados

2. Hipótesis

2.1. Árbol de Búsqueda Binaria convencional

Se espera que para cantidades pequeñas de datos el tiempo de creación de esta estructura sea rápida, al no realizar rebalanceos ni cambios extras al realizar operaciones.

A medida que aumente la cantidad de elementos, se espera que el tiempo de búsqueda e inserción tenga forma logarítmica.

2.2. Árbol AVL

Como este árbol realiza re-balanceos, se espera que el tiempo de creación tome más tiempo que el convencional.

Se espera que para una mayor cantidad de elementos, se comporte mejor que el anterior para todas las operaciones, al encontrarse más comprimido debido a los re-balanceos realizados (va a tener menor altura que el anterior, lo que significa un mejor peor caso).

3. Diseño e implementación

Se desarrollaron todas las estructuras de datos en C. Esta decisión está dada por familiaridad con el lenguaje y por la facilidad para el cálculo del tamaño total de las estructuras. Los árboles estudiados fueron implementados como estructuras, siguiendo una misma firma para el llamado a las operaciones estudiadas. De esta forma todos los métodos pueden seguir la misma batería de tests.

Para el manejo de strings, se trabajó con una tabla de dos entradas. La primera tabla es un único arreglo de caracteres que contiene todo el universo estudiado. La segunda tabla es un arreglo que indica el índice en el cuál inicia cada string. De esta forma los nodos sólo hacen referencia a los strings almacenados. Con este método nos aseguramos que las operaciones estudiadas no consideren el tiempo de aloación o desecho de memoria reservada para strings. A pesar de esto, el tamaño calculado de los nodos si considera el tamaño del string almacenado. Esta decisión fue tomada para facilitar la aloación y desecho de memoria para los strings.

Con el fin de generar los universos de datos, se trabajó con dos script auxiliares en Phyton. El primer script genera archivos de texto con listas de palabras aleatorias de tamaño uniforme. Las palabras generadas tienen un tamaño de 2^4 a 2^8 caracteres y el archivo consta de 2^{15} palabras.

El segundo script toma un archivo de texto plano cualquiera y genera un archivo *pto*¹ que es utilizable por la tarea. Un archivo pto es un archivo binario que contiene la tabla de strings y la tabla de índices, además de algunos datos para facilitar la lectura del archivo mismo. El texto de input es filtrado, eliminando caracteres extraños y el archivo pto es generado.

El código desarrollado se encuentra en el siguiente repositorio

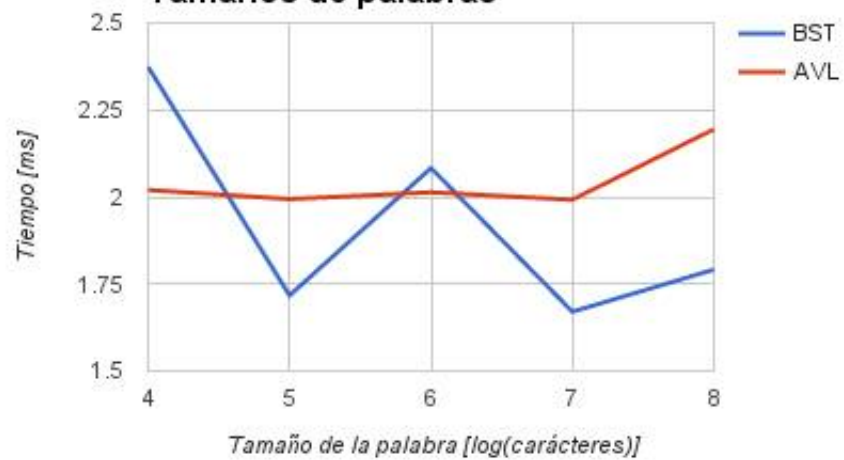
<https://github.com/Pato285/tareas-algoritmos/>

4. Experimentos y resultados

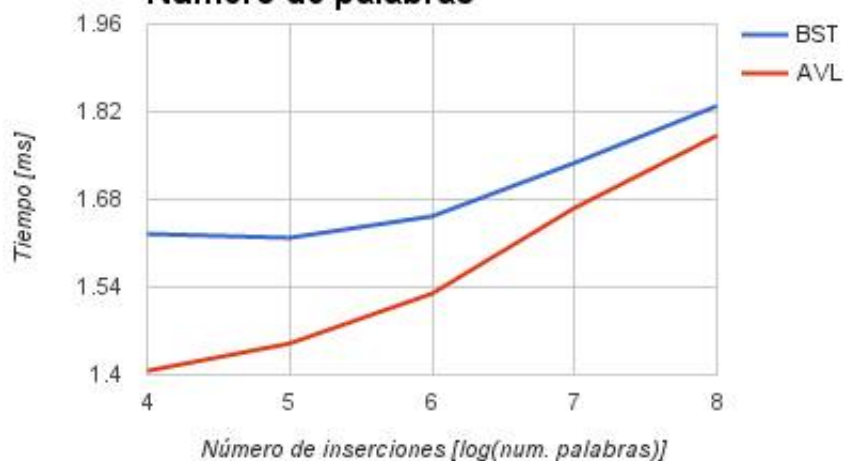
Se ejecutaron las pruebas mencionadas, obteniendo para cada operación el tiempo que tarda por tamaño de las palabras y por número de palabras, cuyos resultados se pueden observar en los siguientes gráficos

¹El formado pto es llamado *Potato file*.

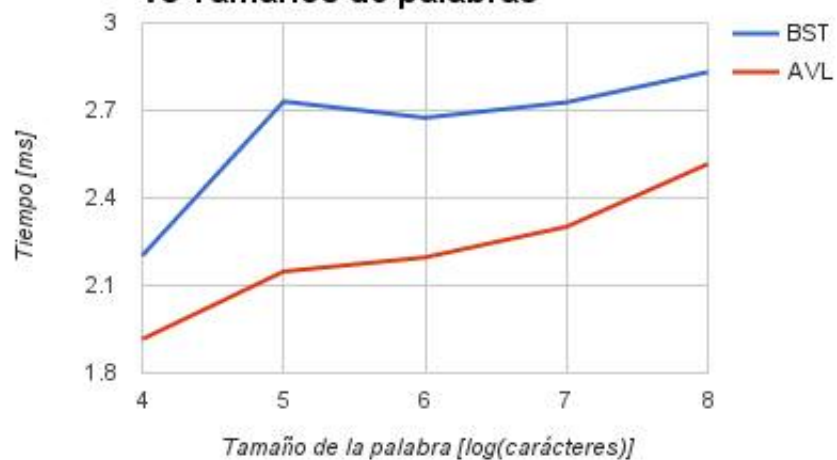
**Tiempos de inserción promedio vs
Tamaños de palabras**



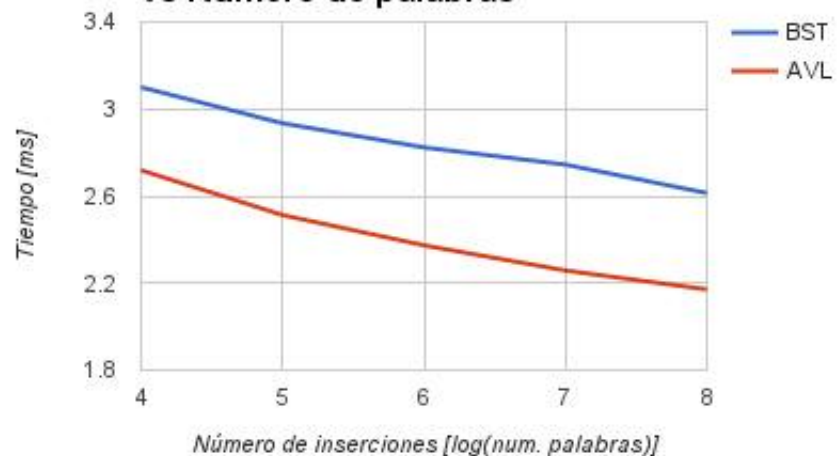
**Tiempos de insertado promedio vs
Número de palabras**



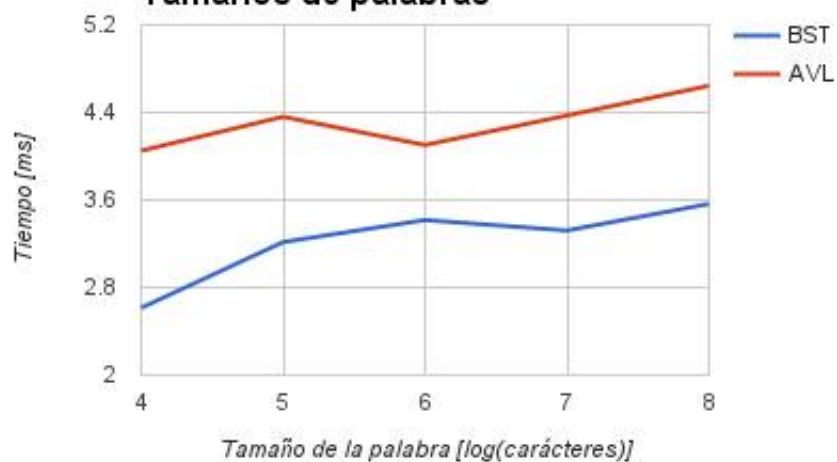
**Tiempos de búsqueda promedio
vs Tamaños de palabras**



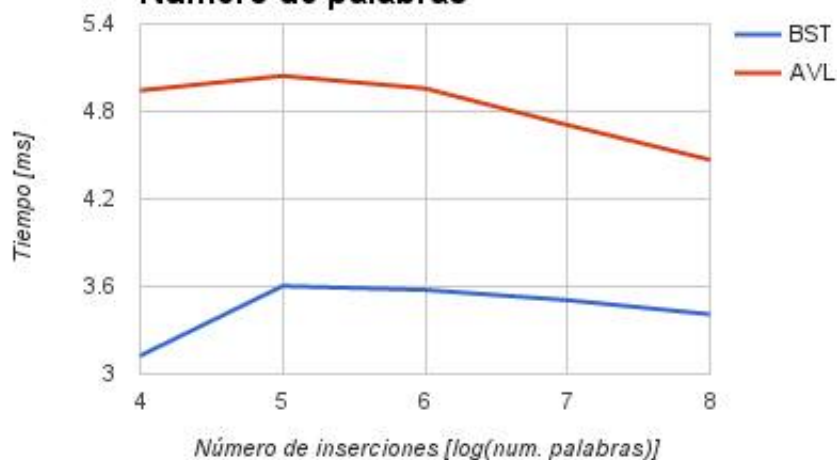
**Tiempos de búsqueda promedio
vs Número de palabras**



**Tiempos de borrado promedio vs
Tamaños de palabras**



**Tiempos de borrado promedio vs
Número de palabras**



5. Análisis y Conclusiones

Con los datos anteriores, se puede apreciar que los árboles AVL superan a los árboles de búsqueda binaria convencional en todas las operaciones (salvo algunos casos de inserción con tamaños específicos de palabras), con esto se puede inferir que el tiempo gastado en los re-balanceos es mucho menor que el tiempo gastado en realizar las operaciones en árboles no balanceados.

6. Anexos

Tiempos Promedio vs Largo de palabras

Word Length [log(char)]	BST Size[Bytes]	BST INSRT Time[ms]	BST SRCH Time[ms]	BST DEL Time[ms]	AVL Size[Bytes]	AVL INSRT Time[ms]	AVL SRCH Time[ms]	AVL DEL Time[ms]
4	321565.0909	2.373696091	2.202165	2.613647727	321565.0909	2.0201	1.915943455	4.050276364
5	512122.1818	1.717207455	2.730613091	3.215207182	512122.1818	1.994040909	2.148024091	4.360504273
6	893236.3636	2.083007818	2.675012727	3.416753727	893236.3636	2.013363909	2.196400091	4.102538909
7	1655464.727	1.670363091	2.727791636	3.320845182	1655464.727	1.992112545	2.301194636	4.372367091
8	3179921.455	1.791365182	2.831529	3.564231182	3179921.455	2.194358091	2.516609909	4.644326182

Tiempos Promedio vs Número de palabras

Number of Words [log(N)]	BST Size[Bytes]	BST INSRT Time[ms]	BST SRCH Time[ms]	BST DEL Time[ms]	AVL Size[Bytes]	AVL INSRT Time[ms]	AVL SRCH Time[ms]	AVL DEL Time[ms]
4	7052.8	1.625	3.1	3.125	7052.8	1.40625	2.71875	4.94375
5	14105.6	1.61875	2.934375	3.603125	14105.6	1.45	2.5125	5.04375
6	28211.2	1.653125	2.8234372	3.5765624	28211.2	1.5296874	2.3734376	4.9578126
7	56422.4	1.7382812	2.7445314	3.50625	56422.4	1.6664062	2.2578124	4.7070312
8	112844.8	1.829297	2.6136718	3.4093752	112844.8	1.7820312	2.170703	4.4683594

Tiempos Promedio de ejecución y Tamaño de estructura en detalle

Word Length [log(char)]	Number of Words [log(N)]	BST Size[Bytes]	BST INSRT Time[ms]	BST SRCH Time[ms]	BST DEL Time[ms]	AVL Size[Bytes]	AVL INSRT Time[ms]	AVL SRCH Time[ms]	AVL DEL Time[ms]
4	4	1728	2.5625	2.34375	2.5625	1728	1.375	2.15625	4.28125
4	5	3456	2.453125	2.25	2.78125	3456	1.390625	2.046875	4.953125
4	6	6912	2.4375	2.445312	2.648438	6912	1.453125	1.953125	4.992188
4	7	13824	2.621094	2.480469	2.679688	13824	1.746094	1.890625	4.339844
4	8	27648	2.671875	2.34375	2.619141	27648	1.816406	1.859375	4.349609
4	9	55296	2.530273	2.189453	2.688477	55296	1.972656	1.821289	3.898438
4	10	110592	2.213379	2.081055	2.568848	110592	2.084961	1.922852	3.959961
4	11	221184	2.077637	1.999756	2.584229	221184	2.203857	1.86792	3.642578
4	12	442368	2.150757	2.014282	2.542603	442368	2.407959	1.819702	3.656616
4	13	884736	2.163757	2.017883	2.555603	884736	2.764893	1.858459	3.379883
4	14	1769472	2.22876	2.058105	2.519348	1769472	3.005524	1.878906	3.099548
5	4	2752	1.3125	2.9375	3.0625	2752	1.34375	2.4375	5
5	5	5504	1.296875	2.765625	3.5	5504	1.375	2.328125	4.8125
5	6	11008	1.359375	2.703125	3.484375	11008	1.492188	2.21875	4.765625
5	7	22016	1.441406	2.699219	3.484375	22016	1.5625	2.148438	4.90625
5	8	44032	1.535156	2.691406	3.441406	44032	1.673828	2.09375	4.410156
5	9	88064	1.625977	2.607422	3.286133	88064	1.862305	2.142578	4.066406
5	10	176128	1.717285	2.992676	3.133789	176128	2.049805	2.083496	3.858887
5	11	352256	1.874756	2.992188	3.071533	352256	2.209473	2.000732	4.297852
5	12	704512	2.033325	2.705811	3.004639	704512	2.460205	2.043213	4.436646
5	13	1409024	2.205994	2.519836	2.946411	1409024	2.728227	1.998474	3.940552
5	14	2818048	2.486633	2.421936	2.952118	2818048	3.123169	2.133209	3.470673
6	4	4800	1.78125	3.34375	3.46875	4800	1.40625	2.71875	4.1875
6	5	9600	1.734375	3.140625	4.1875	9600	1.421875	2.53125	4.328125
6	6	19200	1.75	2.945312	4.046875	19200	1.476562	2.4375	4.4375
6	7	38400	1.796875	2.75	3.738281	38400	1.566406	2.269531	4.363281
6	8	76800	1.947266	2.574219	3.648438	76800	1.742188	2.171875	4.220703
6	9	153600	2.05957	2.482422	3.384766	153600	1.818359	2.100586	4.035156
6	10	307200	2.162598	2.483398	3.21875	307200	1.992188	2.003418	4.304199
6	11	614400	2.438721	2.505127	3.188477	614400	2.168701	2.008057	4.368164
6	12	1228800	2.283325	2.423096	3.041138	1228800	2.467285	1.979004	3.929077
6	13	2457600	2.372253	2.378662	2.930786	2457600	2.9151	1.962036	3.651855
6	14	4915200	2.586853	2.398529	2.73053	4915200	3.172089	1.978394	3.302368
7	4	8896	1.15625	3.375	3.0625	8896	1.3125	2.8125	5.25
7	5	17792	1.25	3.203125	3.671875	17792	1.359375	2.59375	5.25
7	6	35584	1.3125	2.984375	3.726562	35584	1.445312	2.460938	5.164062
7	7	71168	1.375	2.902344	3.6875	71168	1.535156	2.414062	4.816406
7	8	142336	1.460938	2.701172	3.578125	142336	1.638672	2.269531	4.554688
7	9	284672	1.552734	2.554688	3.390625	284672	1.798828	2.148438	4.326172
7	10	569344	1.714944	2.51123	3.247559	569344	2.085449	2.107422	4.141113
7	11	1138688	1.794678	2.4646	3.157715	1138688	2.205566	2.056641	3.984375
7	12	2277376	1.96875	2.481934	3.055176	2277376	2.478638	2.253174	3.769775
7	13	4554752	2.200989	2.429565	3.091675	4554752	2.855469	2.124084	3.529297
7	14	9109504	2.587311	2.397675	2.859985	9109504	3.198273	2.072601	3.31015
8	4	17088	1.3125	3.5	3.46875	17088	1.59375	3.46875	6
8	5	34176	1.359375	3.3125	3.875	34176	1.703125	3.0625	5.875
8	6	68352	1.40625	3.039062	3.976562	68352	1.78125	2.796875	5.429688
8	7	136704	1.457031	2.890625	3.941406	136704	1.921875	2.566406	5.109375
8	8	273408	1.53125	2.757812	3.759766	273408	2.039062	2.458984	4.806641
8	9	546816	1.62793	2.648438	3.638672	546816	2.083984	2.326172	4.487305
8	10	1093632	1.798828	2.573242	3.578125	1093632	2.153809	2.230469	4.208008
8	11	2187264	1.934814	2.602783	3.423584	2187264	2.281006	2.218262	4.036377
8	12	4374528	2.151245	2.58606	3.312012	4374528	2.542114	2.170532	3.86731
8	13	8749056	2.422852	2.622314	3.178284	8749056	2.843079	2.161133	3.730103
8	14	17498112	2.702942	2.613983	3.054382	17498112	3.194885	2.222626	3.537781