

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CC4102 - Diseño y Analisis de Algoritmos

Tarea 1: Diccionarios1

Cristian Carreño Medina
Diego Chávez Escobar
Prof. Jeremy Barbay; Aux. Mauricio Quezada

Índice general

1. Presentación	1
1.1. Introducción	1
2. Experimentacion	2
2.1. Marco Teorico	2
2.1.1. Estructuras	2
2.2. Diseño Experimental	3
3. Resultados y Analisis	4
3.1. Resultados	4
3.1.1. Arbol AVL	4
3.1.2. Arbol Rojo-Negro	5
3.1.3. Arbol 2-3	5
3.1.4. Arbol B	6
3.2. Analisis de Resultados	6
4. Conclusiones	7
4.1. Conclusiones	7

Capítulo 1

Presentación

1.1. Introducción

En el presente informe se detallaran los resultados en terminos de eficiencia de cuatro estructuras de datos, se pondran a prueba las funciones de insercion, busqueda y eliminacion de datos, en las siguientes estructuras:

1. Arbol AVL
2. Arbol Rojo-Negro
3. Arbol 2-3
4. Arbol B

Capítulo 2

Experimentacion

2.1. Marco Teorico

2.1.1. Estructuras

Arbol AVL

Estos Arboles están siempre equilibrados de tal modo que para todos los nodos, la altura de la rama izquierda no difiere en más de una unidad de la altura de la rama derecha o viceversa. Gracias a esta forma de equilibrio (o balanceo), la complejidad de una búsqueda en uno de estos árboles se mantiene siempre en orden de complejidad $O(\log n)$. El factor de equilibrio puede ser almacenado directamente en cada nodo o ser computado a partir de las alturas de los subárboles.

Arbol Rojo-Negro

Un árbol rojo-negro es un árbol binario de búsqueda en el que cada nodo tiene un atributo de color cuyo valor es o bien rojo o bien negro, las hojas poseen valor nulos, los hijos de todo nodo rojo es negro, y el valor de nodos negros en el camino es constante independiente del camino, con esto se logra un arbol aproximadamente equilibrado.

Arbol 2-3

Un árbol 2-3 permite que un nodo tenga dos o tres hijos ademas que todas las hojas han de estar al mismo nivel. Esta característica le permite conservar el balanceo tras insertar o borrar elementos, por lo que el algoritmo de búsqueda es casi tan rápido como en un árbol

de búsqueda de altura mínima. Por otro lado, es mucho más fácil de mantenerlo.

Arbol B

Los árboles-B poseen un numero variable de nodos hijos con un maximo predifnido. Cuando se inserta o se elimina un dato de la estructura, la cantidad de nodos hijo varía dentro de un nodo. Para que siga manteniéndose el número de nodos dentro del rango predefinido, los nodos internos se juntan o se parten. Dado que se permite un rango variable de nodos hijo, los árboles-B no necesitan rebalancearse tan frecuentemente como los árboles binarios de búsqueda auto-balanceables, pero por otro lado pueden desperdiciar memoria, porque los nodos no permanecen totalmente ocupados. Los límites superior e inferior en el número de nodos hijo son definidos para cada implementación en particular.

2.2. Diseño Experimental

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ullamcorper mauris vel nulla egestas sit amet lobortis mi fermentum. Donec elit tellus, feugiat non facilisis non, faucibus sit amet dolor. Fusce feugiat imperdiet mi, ut iaculis eros lacinia vel. Phasellus vel erat eget metus iaculis adipiscing et eu nulla. Cras nec ligula et magna tristique tempus quis sed magna. Nullam non orci at est dignissim auctor at at dui. Vestibulum molestie ultricies libero, eget sollicitudin metus venenatis eu.

In quis tempor risus. Mauris dapibus porttitor nisl, nec lobortis velit fermentum in. Sed fermentum suscipit orci, at porta lorem fringilla in. Pellentesque et arcu et risus vehicula imperdiet. Sed in lorem ac turpis aliquet laoreet a et ante. Nunc ac libero quis diam semper accumsan. Proin semper lectus sed neque volutpat eget scelerisque dolor placerat.

Capítulo 3

Resultados y Analisis

3.1. Resultados

3.1.1. Arbol AVL

Prueba k=3

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=5

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=7

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

3.1.2. Arbol Rojo-Negro

Prueba k=3

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=5

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=7

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

3.1.3. Arbol 2-3

Prueba k=3

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=5

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=7

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

3.1.4. Arbol B

Prueba k=3

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=5

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

Prueba k=7

1. $(i^k d^k i^k)^n = \text{XX ms}$
2. $f^{kn}(d^k i^k d^k)^n = \text{XX ms}$
3. $(d^k i^k d^k)^n = \text{XX ms}$

3.2. Analisis de Resultados

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ullamcorper mauris vel nulla egestas sit amet lobortis mi fermentum. Donec elit tellus, feugiat non facilisis non, faucibus sit amet dolor. Fusce feugiat imperdiet mi, ut iaculis eros lacinia vel. Phasellus vel erat eget metus iaculis adipiscing et eu nulla. Cras nec ligula et magna tristique tempus quis sed magna. Nullam non orci at est dignissim auctor at at dui. Vestibulum molestie ultricies libero, eget sollicitudin metus venenatis eu.

Capítulo 4

Conclusiones

4.1. Conclusiones

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla ullamcorper mauris vel nulla egestas sit amet lobortis mi fermentum. Donec elit tellus, feugiat non facilisis non, faucibus sit amet dolor. Fusce feugiat imperdiet mi, ut iaculis eros lacinia vel. Phasellus vel erat eget metus iaculis adipiscing et eu nulla. Cras nec ligula et magna tristique tempus quis sed magna. Nullam non orci at est dignissim auctor at at dui. Vestibulum molestie ultricies libero, eget sollicitudin metus venenatis eu.

In quis tempor risus. Mauris dapibus porttitor nisl, nec lobortis velit fermentum in. Sed fermentum suscipit orci, at porta lorem fringilla in. Pellentesque et arcu et risus vehicula imperdiet. Sed in lorem ac turpis aliquet laoreet a et ante. Nunc ac libero quis diam semper accumsan. Proin semper lectus sed neque volutpat eget scelerisque dolor placerat.