

CC4102/CC40A/CC53A - Diseño y Análisis de Algoritmos

Auxiliar 6

Prof. Jérémy Barbay, Aux. Mauricio Quezada

11 de Mayo, 2011

Hashing

Sea una tabla (arreglo) de tamaño N y un conjunto de n elementos a insertar.

- Collisions by chaining (Hashing abierto)
 - Utiliza una lista (doblemente) enlazada en cada celda de la tabla. La inserción se produce *al comienzo* de la lista
 - Se define el factor de carga $\lambda = \frac{n}{N}$
 - En el caso promedio:
 - * Buscar y eliminar toman $O(\lambda)$ en promedio si el elemento a insertar no estaba en la tabla. $O(\lambda/2)$ en otro caso.
 - * Insertar toma $O(\lambda)$ en promedio
- Open addressing (Hashing cerrado)
 - Una función de hash h está dada por $h(k) = (h'(k) + f(i)) \bmod N$
 - Por cada colisión, se va buscando en sucesivas posiciones:
 - * Linear probing: $f(i) = i$
 - * Quadratic probing $f(i) = c_1 i^2 + c_2 i$
 - * Double hashing $f(i) = i h''(k)$ (otra función de hash)
- Perfect hashing
 - Sólo requiere $O(1)$ accesos en el peor caso, a diferencia de los anteriores
 - Utiliza una tabla de hash secundaria (Cormen 3th ed. cap. 11.5)
- Universal hashing
 - Suponga que un adversario, conociendo la función de hash, puede escoger los datos a ser insertados. Ya que $n < N$, siempre existirá una secuencia que produzca colisiones.
 - Una familia de funciones de hash $\mathcal{H}$ se dice universal, si para cada par de elementos l, k en el dominio de los valores, el número de funciones $h \in \mathcal{H}$ tal que $h(l) = h(k)$ no excede $\frac{|\mathcal{H}|}{N}$.
- Extendible hashing
 - Si la tabla no cabe en memoria, usar un árbol B^+ para almacenar los valores

Preguntas

1. ¿Qué problemas tiene linear probing? ¿quadratic probing? ¿y double hashing?
2. Dado el tamaño de página $B = 3$, muestre cómo se construye una tabla de hash según el esquema Extendible Hashing con la secuencia

00001, 10000, 10001, 01001, 01111, 00000, 10010, 11111, 10111

Análisis Amortizado

1. ¿Cuál es el costo amortizado de n inserciones en un vector (inicialmente de tamaño 1), que duplica su tamaño cada vez que se llena?
2. Muestre que el costo amortizado de n incrementos de un contador binario es $O(1)$ por inserción. Utilizando: el método agregado, de conteo y potencial.
3. Pruebe, utilizando análisis amortizado, que la heurística *Move To Front* (MTF) rinde según un factor de 4 con respecto a *cualquier* otro algoritmo de búsqueda en un arreglo desordenado.