

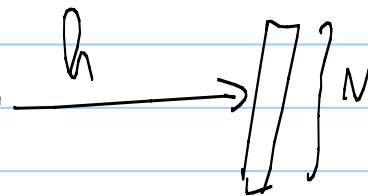
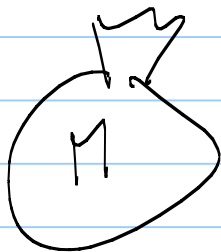
Hashing

Note Title

10/29/2009

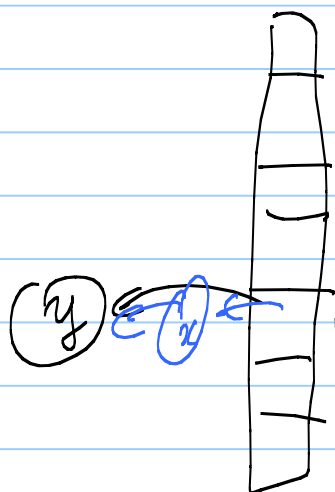
① Basics

* hash function = resumen
 $f: [0, n-1] \rightarrow [0, N-1]$



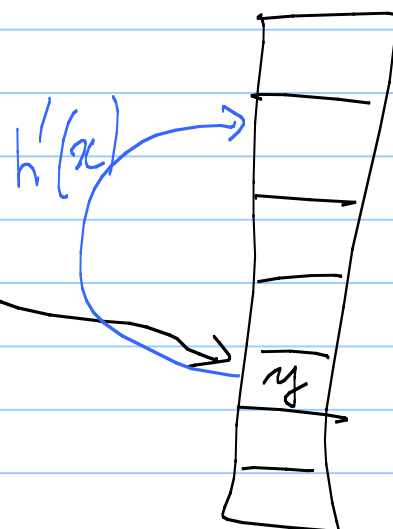
$\Rightarrow$ Operaciones en $O(1)$ en promedio
pero $O(n)$ en peor caso
(con algunas excepciones)

* Open Hashing
Hash Abierto



x
 $\downarrow$
 $h(x)$

* Closed Hashing
Hash Cerrado



② En memoria Externa (con Cache)

Extendible Hashing

- Paralelo con "Arbol B"

- B arbol de altura 2.

- una raíz "directorio"
- hojas con los datos

- una sola valor de hash para cada llave
bits mas significativos $\Rightarrow$ directorio
menos $\Rightarrow$ hoja

B = cantidad de elementos en una página

$h :$ $\rightarrow [0, 2^k - 1]$

$D :$ profundidad global {

$d_l :$ profundidad de una hoja l

$\Rightarrow$ valores de h en l tienen d_l bits en comun.

$\Rightarrow$ hay 2^{D-d_l} punteros a la hoja l . (de peso o mas significativos)

Ejemplo $B=4, k=6, D=2$

$D=2$
↔

00
01
10
11

2 0 0 0 1 0 0
0 0 1 0 0 0
0 0 1 0 1 1
0 0 1 1 0 0 } $B=4$

2 0 1 0 1 0 1
0 1 1 1 0 0

1 1 0 0 1 0 0
1 0 1 1 0 1
1 1 0 0 0 1
1 1 1 1 0 0

Operadores

* Busqueda:

$$\begin{cases} h(x) \\ i \leftarrow h(x) \oplus 2^p - 1 \\ \text{seguir } P[i] \\ \text{escanear la pagina} \end{cases}$$

Ej: Buscar 110001

* Insertar idem que buscar (7) 3 casos

① hay espacio en la hoja

② la hoja es llena, hay que separarla

si $d_e < D$, - divide la hoja

- inserta x

- actualiza los punteros

③ si $d_e = D$ - $D \leftarrow D+1$

- Copia el directorio en nuevo arreglo

- idem que ②

* Remove

+ Como Buscar (+) remove el elemento

+ intenta de fusionar con "pareja"

↳ cual puntero en el directorio
tiene $d-1$ bits de peso más sign.
en común + están al
mismo nivel de

+ p o demos tener mas que una fusion

+ Reduce 0 si possible

Complejidad
Busqueda

- supongamos que el directorio
siempre es en memoria

$\Rightarrow$ 1 acceso al disco

Insertar } mejor caso $\Rightarrow$ 1 acceso al disco
Borrar } peor caso : ∞ cota en

En promedio, $\frac{n}{B \ln 2} \approx 1.44 \frac{n}{B}$ paginas para n claves

cada pagina es

69% llena

Características

- garantiza un acceso en búsqueda
- directorio tiene que quedarse en memoria
- no hay control sobre la utilización del espacio en disco
- no hay garantías sobre el porcentaje de cada página utilizado **en el peor caso**.
(en promedio sí)

② Hashing Linear.