Diccionarios en Memoria Secundaria

Note Title 8/31/2010

I B- arboles

① (2,3) arboles
- no son binarios (grado 2 o 3)
- de busqueda
- balanceados de manera PERFECTA

- se busca - cada nodo (k_1, k_2) $O(\lg n)$
- se inserta (buscar la hoja, dividir nodos hasta encontrar espacio)
- se remueve (

② $(d, 2d-1)$ arboles

Ve en ③ B-arboles

- d a $2d-1$ hijos
  - $\Rightarrow$ $d-1$ a $2d-2$ llaves por nodo.
  - $\Rightarrow$ con la excepcion de la raiz
- buscar
  - busqueda a dentro del nodo (¿lineal?)
  - sigue recursivamente en sobre arbol
- inserta
  - busque, si $(k_1, \ldots, k_{2d-2})$
    - $\Rightarrow$ dividir $(k_1, \ldots, k_{d-1}), k_d, (k_{d+1}, \ldots, k_{2d-1})$

- remover

- buscar
- si en el ultimo nivel
  - si = d-1 llaves y no la raiz
    - ② bajar una llave del nodo superior y seguir recursivamente.
    - o
    - ① tomar prestado a un hermano consecutivo
  - si > d-1 llaves o la raiz
    nada que hacer
- otramente, si en un nivel entremedio
  reemplazar por el predecesor
  (o sucesor)

③ B-arboles

- un $(d, 2d-1)$ arbol tal que
  - un nodo queda en $O(1)$ paginas del cache
  - en cada nodo se puede usar cualquier estructura de datos de diccionarios

Propriedades

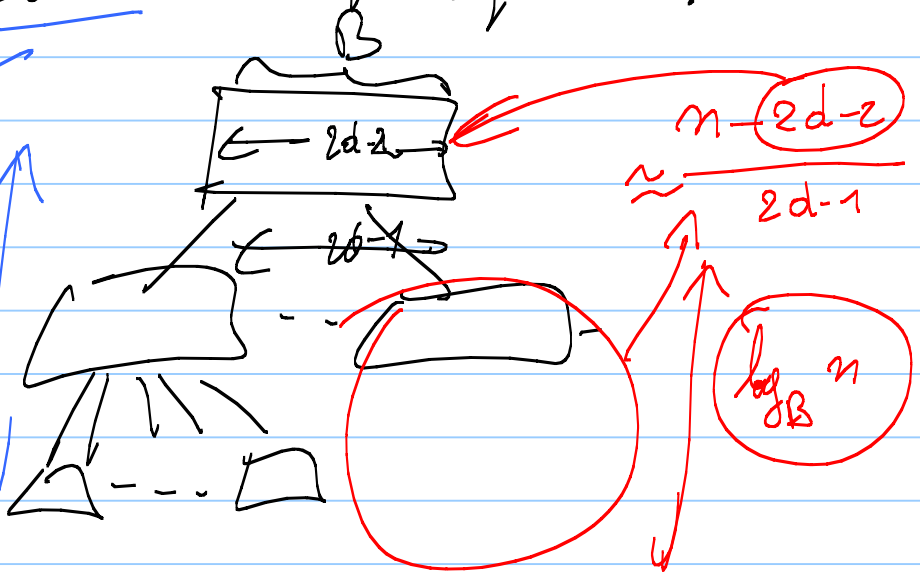
Cuanto cuesta una remoción / inserción / búsqueda?

accesos al coche (n, B)

$\Downarrow$

$2d-2$

$\frac{\log n}{\log B}$



Applicaciones

- Base de Datos
  - B^*
  - B^+

- usualmente se duplican las llaves
  - una occurencia en el arbol
  - una en la hoja.

Van Emde Boas

Concepto: $(\sqrt{n}/2, \sqrt{n}-1)$ arbol

$\rightsquigarrow$ hasta $\sqrt{n}$ hijos,

cada sobre arbol con $\sqrt{n}$ hojas

$\Rightarrow$ Altura en $O(\lg \lg n)$

Historia: ① Definidos por el equipo de "Van Emde Boas", donde cada nodo tiene una tabla de hash $\Rightarrow$ Poco Practica

② Aplicado a paralelismo llego una sorpresa: buen rendimiento por todos caches

Cache Oblivious $\Leftarrow$

Finger Search Tree

~> similar a busqueda Doblada, pero en arboles de busqueda.
→ busca en tiempo $O(\lg p)$ donde p es la distancia al ultimo elemento buscado.

- inicia la busqueda en una hoja, llamada "finger" o dedo.

- sube el arbol hasta encontrar un sobre arbol que contiene la llave (si necesario, seguir puntero horizontal al "primo") ~> $O(\lg p)$
- bajar el sobre arbol de manera regular ~> $O(\lg p)$

Nota Van Emde Boas (static)

y

Finger Search Trees

en Tarea 2!!!