

CC4102/CC40A/CC53A - Diseño y Análisis de Algoritmos

Auxiliar 3

Prof. Jérémy Barbay, Aux. Mauricio Quezada

13 de Abril, 2011

1 B-Árboles

1. Un B-Árbol T es un árbol con las siguientes propiedades:

- Cada nodo x posee
 - $n[x]$ el número de llaves en x
 - las $n[x]$ llaves, en orden no-decreciente: $key_1[x] \leq \dots \leq key_{n[x]}[x]$
 - $leaf[x]$ un valor booleano que indica si x es un nodo interno o no
- Cada nodo interno además posee $n[x] + 1$ punteros $c_1[x], \dots, c_{n[x]+1}[x]$ a sus hijos. Las hojas no tienen hijos, por lo que sus c_i están indefinidos
- Las llaves $key_i[x]$ separan los rangos de llaves almacenadas en cada subárbol. Si k_i es una llave almacenada en el subárbol con raíz $c_i[x]$, entonces

$$k_1 \leq key_1[x] \leq k_2 \leq key_2[x] \leq \dots \leq key_{n[x]}[x] \leq k_{n[x]+1}$$

- Todas las hojas tienen la misma profundidad, la cual es la altura del árbol h
- El grado de T , $t \geq 2$ es tal que
 - cada nodo distinto de la raíz debe tener al menos $t - 1$ llaves.
 - cada nodo puede tener a lo más $2t - 1$ llaves

2. Pruebe que, si $n \geq 1$, para cualquier B-Árbol de n llaves, altura h y grado $t \geq 2$, $h \leq \log_t \frac{n+1}{2}$

3. De el algoritmo de inserción en un B-Árbol

2 van Emde Boas Trees

1. Nos gustaria crear una estructura de datos con las siguientes propiedades

- Mantener un conjunto dinámico S de tamaño n de un dominio $\mathcal{U} = \{0, \dots, u-1\}$, $|\mathcal{U}| = u$
- Que soporte las siguientes operaciones:
 - (a) **Insert** ($x \in \mathcal{U}, x \notin S$): Agregar x a S
 - (b) **Delete** ($x \in S$): Eliminar x de S
 - (c) **Succ** ($x \in \mathcal{U}$): Encontrar el menor elemento $z \in S$ tal que $z > x$
 - (d) **Pred** ($x \in \mathcal{U}$): Encontrar el mayor elemento $z \in S$ tal que $z < x$

Describa un van Emde Boas Tree (o vEB Priority Queue)

2. Muestre que un van Emde Boas Tree soporta estas operaciones en tiempo $\mathcal{O}(\lg \lg u)$. ¿En qué casos las operaciones funcionan en $\mathcal{O}(\lg \lg n)$?
3. Pruebe que esta estructura utiliza espacio $\Theta(u)$
4. (Propuesto) Determine un algoritmo **Find** ($x \in \mathcal{U}$) que busca x en S en tiempo $\mathcal{O}(\lg \lg u)$