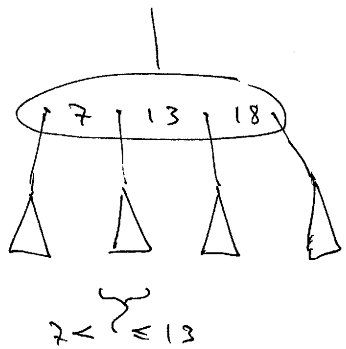
(a, b) - trees

Definition

Node:



Multiway search trees where:

+ All leaves on same level.
+ Leaves contain from a to b elements
+ Internal nodes ($\neq$ root) have degrees from a to b.
+ Root has degree from 2 to b.

Here, a, b are integers with $a \geq 2$, $b \geq 2a-1$

B-trees are (a,b)-trees with $a, b \in \Theta(B)$

Last time: (Predecessor) Search in $O(\log_B(N))$ I/O's

Range Search in $O(\log_B(N) + \frac{|\text{output}|}{B})$ I/O's

$\Theta(\log_B(N))$ search bound is optimal in comparison based I/O-model.

Today: Updates

We assume leaf-oriented trees ("B^+-trees" in some DB-books): elements all reside in leafs, leafs are linked together to form a sorted list of blocks ($\Rightarrow$ Range Queries are even simpler than discussed last time), internal nodes contain (copies of) element keys that just guide the search.

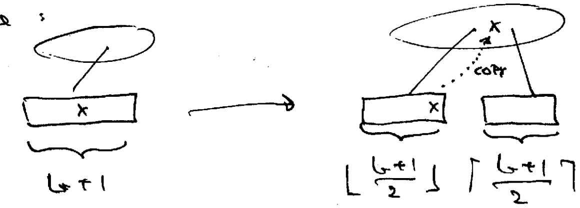
Update = search for right leaf + ins/del element in leaf + possible rebalance.

Rebalance:

Insertion may generate overflow = node of size b+1

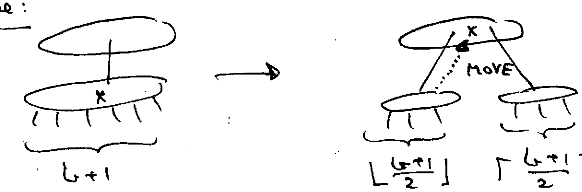
$\boxed{\text{Split}}$ node:

Leaf:



(x = copy of largest key in left leaf)

Internal Node:



Note : $a \le \left\lfloor \frac{b+1}{2} \right\rfloor \le \left\lceil \frac{b+1}{2} \right\rceil \le b$, since

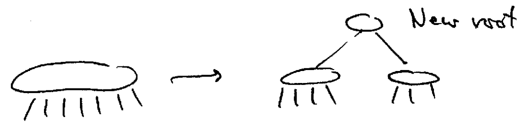
$b \ge 2a-1$ and $b \ge 1$, so new

nodes are legal. <u>But</u> : size of parent grows

by one $\Rightarrow$ possible over flow. May propagate

to root.

$\boxed{\text{Split at root}}$:

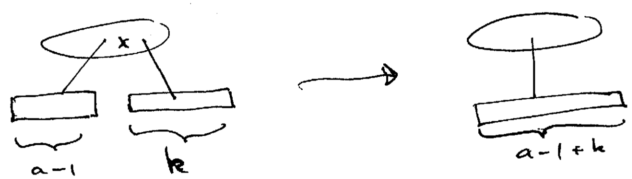


New root legal. Height of tree grows by one.

<u>Deletion</u> may generate underflow = node of size $a-1$.

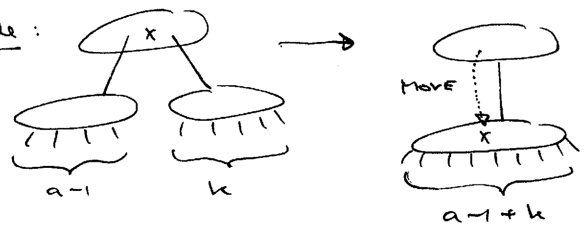
$\boxed{\text{Fuse}}$ node with a sibling

<u>Leaf</u> :



(delete x in top node)

<u>Internal Node</u> :



Note : Fuse ~ split in reverse.

Legal nodes afterward ?

+ Size of parent falls by one $\Rightarrow$ possible underflow. May propagate to root.

+ Size of new node : $a \le k \le b$ always. Choose a threshold t, $2a \le t \le b+1$, and <u>split</u> new node if its size $\ge t$.

Not split $\Rightarrow$ size between $a-1+a > a$ and $t-1 \le b$.

Split $\Rightarrow$ sizes between $\left\lfloor \frac{b+1}{2} \right\rfloor \ge a$

and $\left\lceil \frac{a-1+b}{2} \right\rceil < \left\lceil \frac{2b-1}{2} \right\rceil = b$

So in both cases, nodes at bottom of drawing are legal.

Fuse + Split = $\boxed{\text{Share}}$

Note : A share does not change size of parent $\Rightarrow$ no propagation.

Fuse at children of root of size two will remove root ($\Rightarrow$ tree height decreases one)

Threshold $t = b+1$ in Lars Arge notes.

In many textbooks $t = 2a$

Other issues :

Size of B-trees/(a,b)-trees :

$\leq N/a$ leafs

$\leq \#\text{leafs}/a \leq \frac{N}{a^2}$ nodes on next level

$\leq \frac{N}{a^3}$ nodes on next level again

$\vdots$

$$\text{Total} \leq \frac{N}{a} \cdot \sum_{i=0}^{\infty} \left(\frac{1}{a}\right)^i = \frac{N}{a} \cdot \frac{1}{1 - 1/a} = \frac{N}{a} \frac{a}{a-1}$$

$$\leq 2 \cdot \frac{N}{a} \text{ nodes}$$

For B-trees : $\Theta(N/B)$ blocks.

Build (from N unsorted elements)

1) Repeated insertions : $O(N \cdot \log_B N)$

2) Sort elements and build level-by-level bottom-up (scanning one level while creating the next) :

$$O\Big(\text{Sort}(N) + \text{Scan}\big(\underbrace{\tfrac{2N}{B} \cdot B}_{\text{size of tree}}\big)\Big)$$

$$= O(\text{Sort}(N) + \text{Scan}(N))$$

$$= O(\text{Sort}(N)) = O\left(\frac{N}{B} \cdot \log_{M/B}(N/M)\right)$$

2) is always better.

Comparisons for search :

Binary search in tree nodes : $\log_2(B) \cdot \log_B(N) = \log_2(B) \cdot \frac{\log_2(N)}{\log_2(B)} = \log_2(N)$

Linear —— " —— : $B \cdot \log_B(N)$

$> \log_2(B) \cdot \log_B(N)$.

(optimal in comparison based model)