B tree time complexity

09 October 2020 18:23

Let 'T' be the B-tree of order 'm' and height 'h'. Let $t = \lceil m/2 \rceil$

and let 'n' be the number of keys in 'T'. Then

(i) $2t^{h-1} - 1 \le n \le m^h - 1$

(ii) $\log_m(n+1) \le h \le \log_t\left(\frac{n+1}{t}\right) + 1$

Proof:

First let us find minimum number of nodes in 'T'.

Let us take $m = 5$.

Then $t = \lceil \frac{m}{2} \rceil = 3$

Level	No. of nodes
1	1
2	$2 = 2 \times 3^0 = 2t^0$
3	$6 = 2 \times 3^1 = 2t^1$
4	$18 = 2 \times 3^2 = 2t^2$
⋮	
'h'	$2 \times t^{h-2}$

Total number] $= 2t^0 + 2t^1 + \dots + 2t^{h-2}$

of nodes except root node $\Big\} = 2t + 2t + \cdots$

$= 2(1 + t + t^2 + \cdots t^{h-2})$

$= 2\dfrac{(t^{h-1} - 1)}{(t-1)}$

Total number of keys (except root) $= 2\dfrac{(t^{h-1} - 1)}{\cancel{t-1}} \times \cancel{t-1}$ [∵ Each node has $(t-1)$ keys]

$= 2(t^{h-1} - 1)$

$= 2t^{h-1} - 2$

∴ Total no. of keys including root $\Big\} = n = 2t^{h-1} - 2 + 1$

$n = 2t^{h-1} - 1$

Since 'n' is minimum

$\boxed{n \geq 2t^{h-1} - 1}$ — (1)

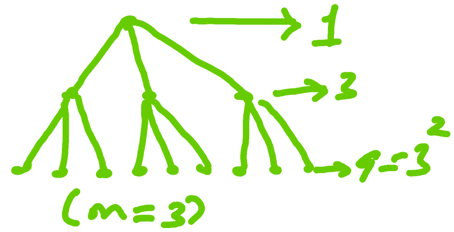
<u>Maximum number of keys</u>

<u>Level</u>	<u>No. of nodes</u>
1	1 (root node)
2	m (∵ each level has atmost 'm' child)

3 $\quad m^2$

$\vdots$

$h \quad m^{h-1}$



$\therefore$ Total no. of nodes $= 1 + m + m^2 + \cdots m^{h-1}$

$$= \frac{m^h - 1}{m-1}$$

Total no. of keys $= \dfrac{m^h - 1}{\cancel{m-1}} \times \cancel{m-1}$ ($\because$ Each node has $m-1$ keys)

$$n = m^h - 1$$

Since 'n' is maximum

$$\boxed{n \le m^h - 1} \text{ --- } (2)$$

From (1) & (2) it follows

$$\boxed{2t^{h-1} - 1 \le n \le m^h - 1}$$

Now,

from (1),

$$2t^{h-1} - 1 \le n$$

$$2t^{h-1} \leq n+1$$

$$t^{h-1} \leq \frac{n+1}{2}$$

Taking log to base 't'

$$\log_t t^{h-1} \leq \log_t \frac{n+1}{2}$$

$$(h-1)\log_t t \leq \log \frac{n+1}{2}$$

$$h-1 \leq \log \frac{n+1}{2}$$

$$\boxed{h \leq \log \frac{n+1}{2} + 1} \quad \text{--- (3)}$$

From (2)

$$m^h - 1 \geq n$$

$$m^h \geq n+1$$

Taking log to base 'm'

$$\boxed{h \geq \log_m (n+1)} \quad \text{--- (4)}$$

From (3) & (4),

$$\boxed{\log_m (n+1) \leq h \leq \log_t \left(\frac{n+1}{2}\right) + 1}$$

Hence we find that height of B tree doesn't exceeds log n.

$\therefore$ Insert, search, delete operations of B-Tree is $O(\log n)$.