Proof of B-Tree's height

Abdul Fatir Ansari

Prove that the height of a B-Tree with n keys is $O(\log n)$.

Let's begin by finding the minimum number of keys a B-Tree of height h should have. Assume that the branching factor is b.

- (height 0) Root will have at least 1 key (Remember that root must have 1 to $2b-1$ keys). Total nodes at this height: 1.
- (height 1) Since the root has at least 1 key, it will have 2 child nodes. Total nodes at this height: 2.
- (height 2) Since the nodes at height 1 have $b-1$ keys (Remember that non-root nodes must have $b-1$ to $2b-1$ keys), they will have b children each. Total nodes at this height: $2b$.
- (height 3) Every one of the $2b$ nodes at height 2 will have b children each. Total nodes at this height: $2b^2$.
- $\vdots$
- (height h) Total nodes at this height: $2b^{h-1}$.

Apart from the root node, all other nodes are required to have at least $b-1$ keys. So, the total minimum number of keys a tree of height h must have is:

$$\begin{aligned}
&= \underbrace{1}_{\text{for root node}} + \underbrace{(2 + 2b + 2b^2 + \cdots + 2b^{h-1})(b-1)}_{\text{for non-root nodes}} \\
&= 1 + 2(b-1)(1 + b + b^2 + \cdots + b^{h-1}) \\
&= 1 + 2(b-1)\frac{(b^h - 1)}{(b-1)} \qquad \text{(Geometric Progression)} \\
&= 2b^h - 1
\end{aligned}$$

The minimum number of keys $\leq n$, i.e.,

$$2b^h - 1 \leq n$$

$$
\begin{aligned}
b^h &\leq \frac{n+1}{2} \\
\log_b b^h &\leq \log_b \left(\frac{n+1}{2} \right) \\
h &\leq \log_b \left(\frac{n+1}{2} \right) \\
h &\leq \frac{\log \left(\frac{n+1}{2} \right)}{\log b}
\end{aligned}
$$

Ignoring constant terms we have shown that the height of a B-Tree with n keys is $O(\log n)$.