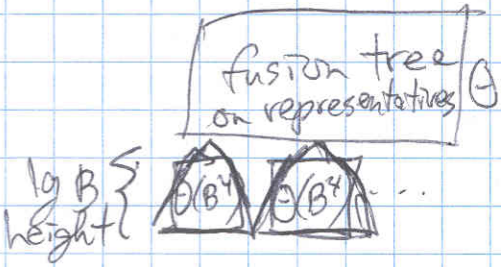
Updates

- usual B-tree.
- cost to rebuild (touch) a node is $O(B^4)$
- use indirection to slow updates by $\times \Theta(B^4)$.



= Cost @ bottom = $\Theta(\lg B) = \Theta(\lg \lg n)$ ~ negligible

- Fusion tree updates only every $O(B^4)$ real updates (am.)
- B-tree touches $O(1)$ nodes amortized / update

$\Rightarrow O(1)$ amortized.

$O(\lg \lg n)$ overall for update (excluding search)

Speed up to $O(\sqrt{\lg n})$:

if $\lg \lg u \le \sqrt{\lg n}$
i.e. $(\lg \lg u)^2 \le \lg n$
i.e. $n \ge 2^{(\lg \lg u)^2}$
then use vEB/y-fast.
else: $n \le 2^{(\lg \lg u)^2}$

$\Rightarrow 2^{\frac{1}{5}\sqrt{\lg n}} \le 2^{\frac{1}{5}\lg \lg u} = (\lg u)^{1/5}$.

so can use $B = 2^{\frac{1}{5}\sqrt{\lg n}}$.

cost of fusion tree search $= \log_B n = \frac{\lg n}{\Theta(\sqrt{\lg n})} = \Theta(\sqrt{\lg n})$

cost at bottom $= \Theta(\lg B^4) = \Theta(\lg B) = \Theta(\sqrt{\lg n})$.

& balance. tah dah!