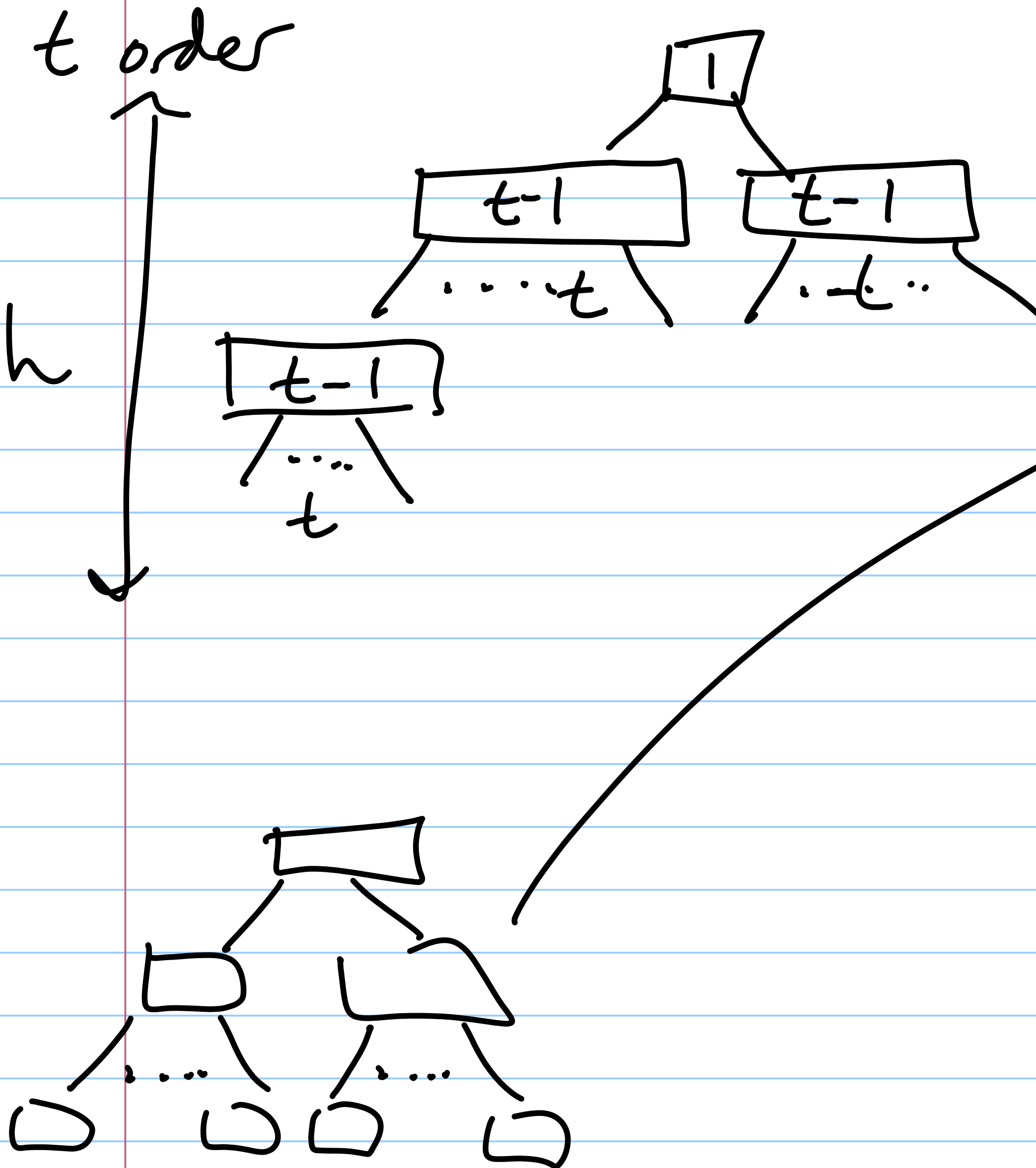
Analysis of height

suppose the B-tree has height h

What is the min # of elements it might hold?

$n \geq F(h)$ solve for h $h \leq f(n)$



	# elements
level 0	1
	$2(t-1)$
	$\underbrace{2t}_{\#\text{ nodes}} \cdot (t-1)$
	$2t^2(t-1)$
	$2t^3(t-1)$
	$\vdots$
level h-1	$2t^{h-1}(t-1)$

$$n \geq 1 + \underbrace{2(t-1)+2t(t-1)+2t^{2}(t-1)+\cdots+2t^{h-1}(t-1)}$$

$$n \geq 1 + 2(t-1)\underbrace{\left(1+t+t^{2}+\cdots+t^{h-1}\right)}$$

$$n \geq 1 + 2\cancel{(t-1)}\,\frac{(t^{h}-1)}{\cancel{(t-1)}}$$

$$n \geq 1 + 2(t^{h}-1)$$

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

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

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

$r \neq 1$

$$\boxed{\sum_{i=0}^{x} r^{i} = \frac{r^{x+1}-1}{r-1}}$$

$$2t^h \le n+1$$

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

$$\underbrace{h \le \log_t\left(\frac{n+1}{2}\right)}$$

maximum height of B-tree with n elements

$$\log_t\left(\frac{n+1}{2}\right)$$

$$= \log_t(n+1) - \log_t 2$$

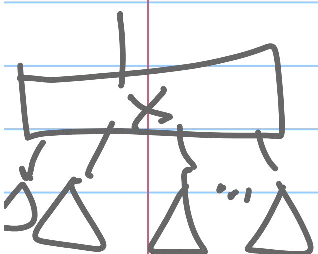
$$\boxed{\approx \log_t n}$$

Cost for insertion

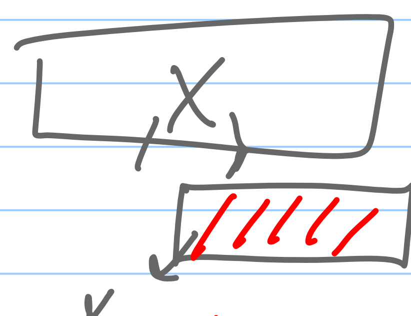
$$O\Big(\underbrace{\boxed{\log_t n}}_{\text{\# nodes on path down}} \cdot \underbrace{\log_2(2t-1)}_{\text{time per node (sorted array)}}\Big) = O(\log_2 n)$$

max # of page faults

Overview of B-tree Deletion



Like binary search tree, for removing element in an internal node, then replace x by its successor & remove the successor



from marked node take leftmost child until reach leaf & succ. leftmost element

hold this in memory until successor is found to replace it