Amortized analysis (reminder):

Claiming that rebalancing is amortized $O(f(n))$ in (a,b)-trees means that we claim that: Starting with an empty tree, then after k updates (and any number of searches, since these do not generate any rebalancing) the total number of rebalancing operations [SPLITS, FUSES, SHARES, ROOT OPS.] is $O(k \cdot f(n))$.

Amortized Analysis of (a,b)-trees

Thm For empty start tree, the amortized number of rebalancing operations in an (a,b)-tree is:

1) $\Theta(\log_a N)$ for $2a-1 = b$

2) $\Theta(1)$ for $2a = b$

3) $\Theta(\frac{1}{\varepsilon \cdot a})$ for $(2+\varepsilon)a = b$ $(\varepsilon > 0)$.

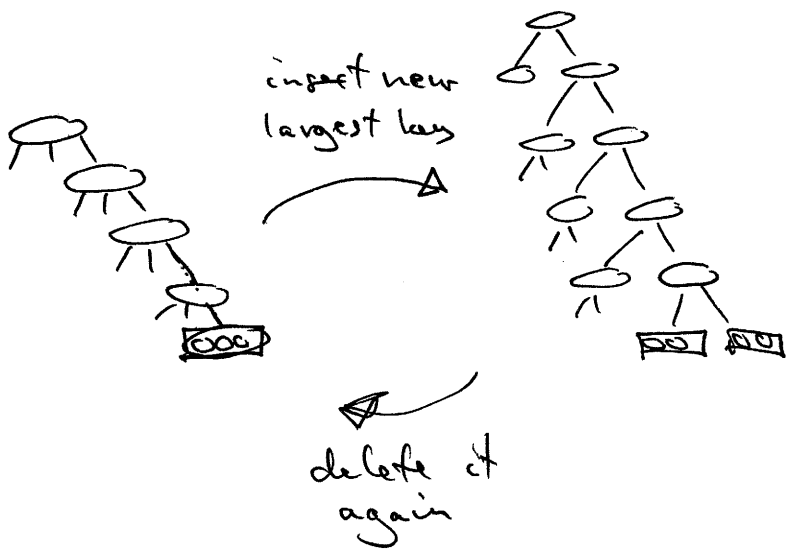
Note: i) the threshold phenomenon when b grows by one from the minimal value possible $(2a-1)$.

ii) Rebalancing in B-trees can be made amortized $O(\frac{1}{B})$ for $3a = b = B$ (e.g.)

Proof of thm:

1)

For eg. (2,3)-trees:
[Same example applies to all $(a, 2a-1)$-trees]



So rebal. propagates to the root each update in this example.

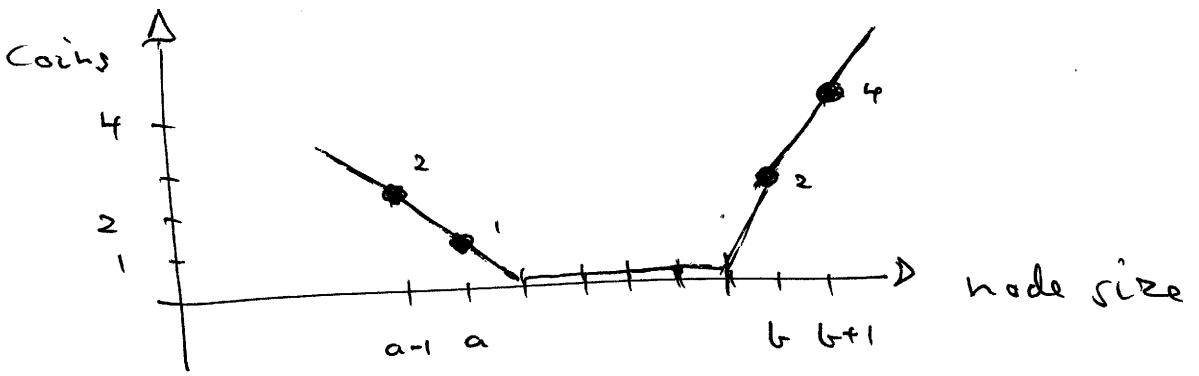
⇒ amortized rebal. is $\Omega(\text{height}) = \Omega(\log_a(N))$.

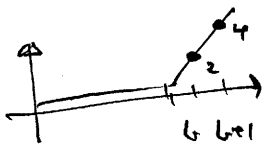
[Rebal. is always $O(\log_a N)$ worst case ⇒ is also $O(\log_a N)$ amortized. So really $\Theta(\,)$]

2) Idea of proof: updates are charged (leave coins in tree) the amortized cost always, then we prove stored coins always can pay the rebalancing done.

> One coin can pay for one rebalancing operation.

Store coins in nodes. Rule for this:



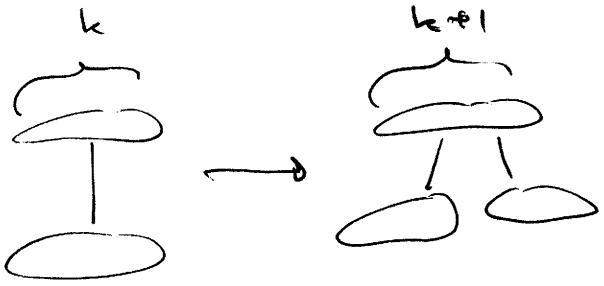
(For root: only right part of graph 

)

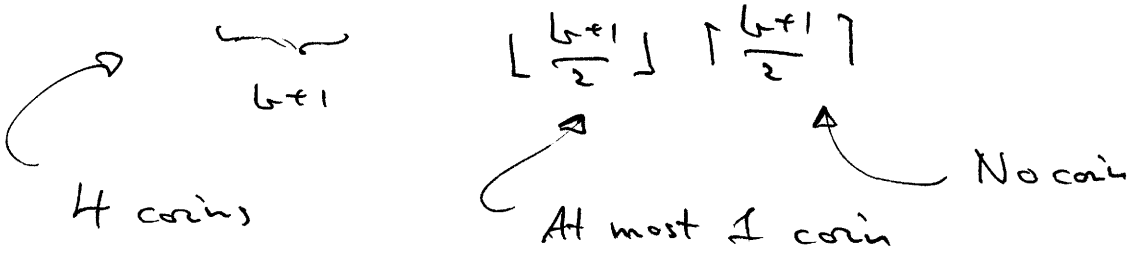
Note: At most one SHARE or ROOT operation per update. So we don't need to bound these by other means.

Coins will pay for all SPLITS and FUSES.

SPLIT :



← Maybe two coins more.



$$\left\lfloor \frac{k+1}{2} \right\rfloor \geq \left\lfloor \frac{2a+1}{2} \right\rfloor \geq a \qquad (k \geq 2a)$$

$$\left\lceil \frac{k+1}{2} \right\rceil \geq \left\lceil \frac{2a+1}{2} \right\rceil = \left\lceil a + \tfrac{1}{2} \right\rceil = a+1$$

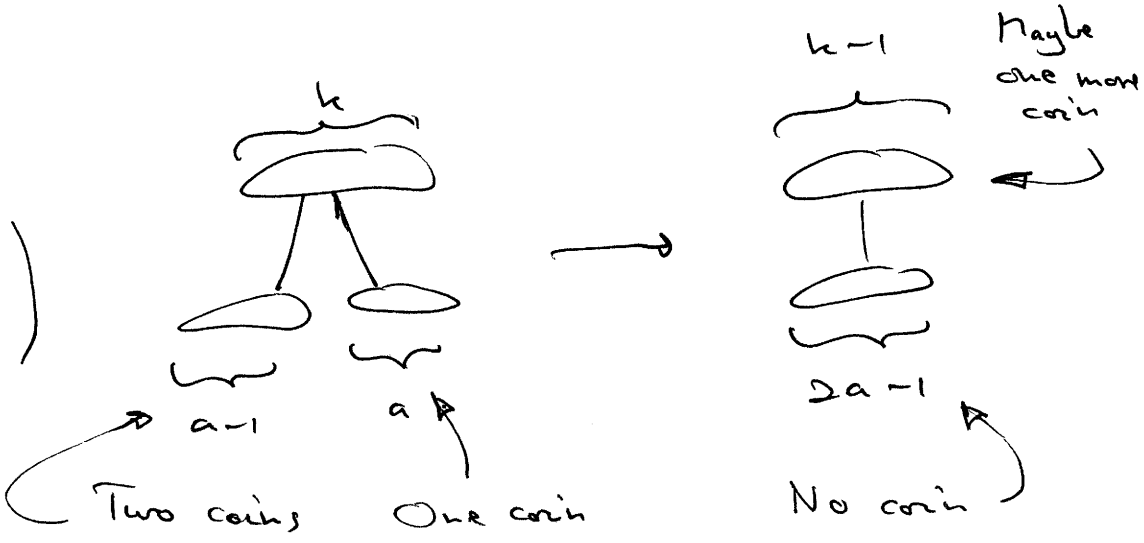
$$\left\lceil \frac{k+1}{2} \right\rceil = \left\lceil k - \frac{k}{2} + \frac{1}{2} \right\rceil \leq \left\lceil k - 2 + \frac{1}{2} \right\rceil \leq k-1$$

$(k \geq 2a \geq 2 \cdot 2)$

At least one coin is released to pay for the SPLIT

FUSE :

(Assume split threshold $t = 2a$)

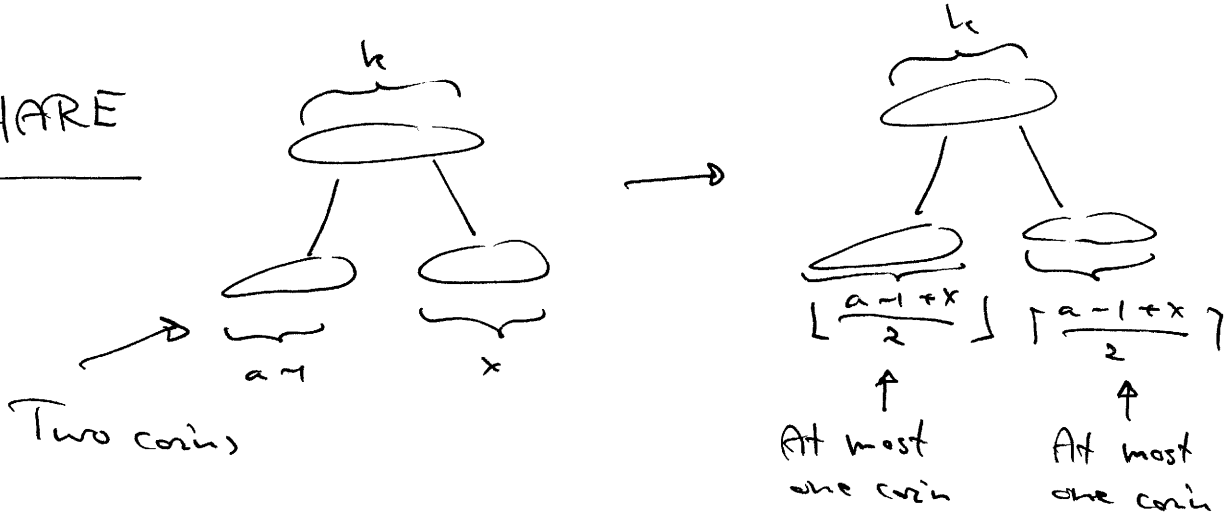


$$2a-1 = a+(a-1) \geq a+1 \qquad (a \geq 2)$$

$$2a-1 \leq b-1 \qquad (b \geq 2a)$$

At least two coins are released to pay for the FUSE.

SHARE



$$a+1 \leq x \leq b$$

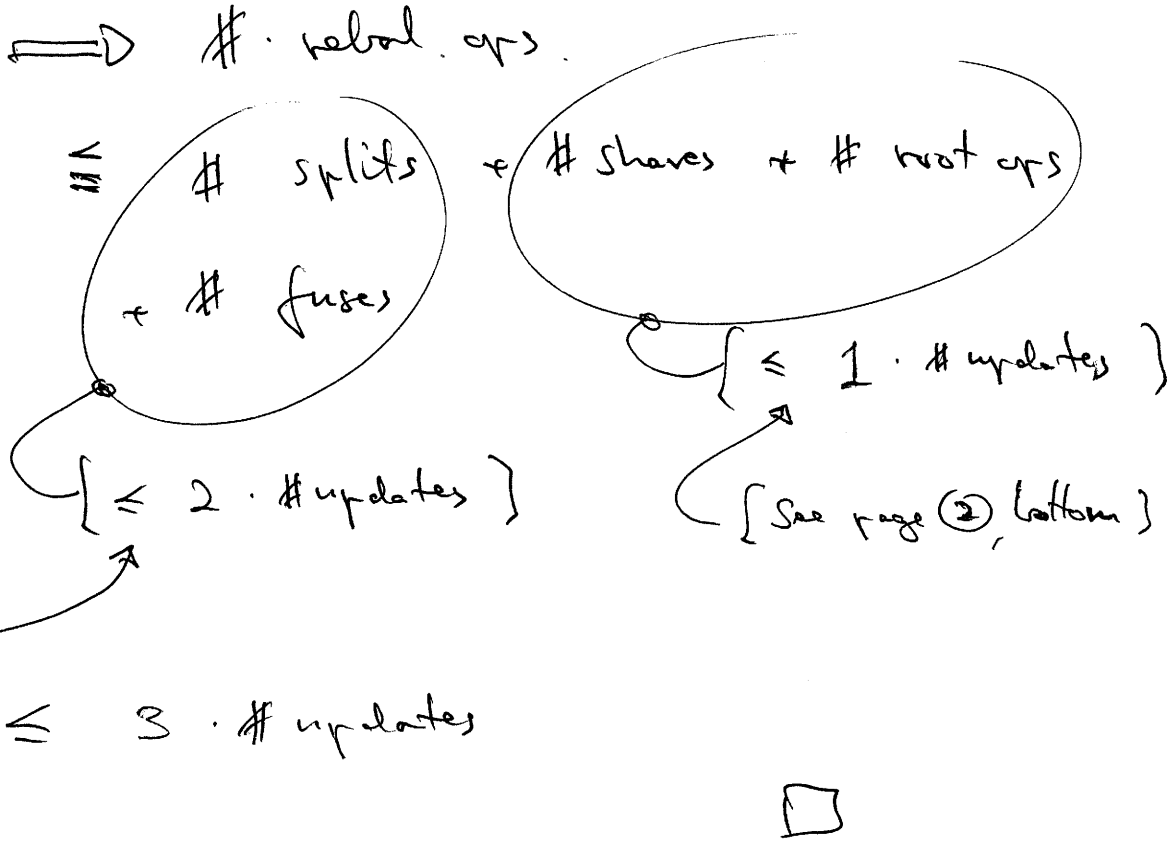
$$\left\lfloor \frac{a-1+x}{2} \right\rfloor \geq \left\lfloor \frac{a-1+(a+1)}{2} \right\rfloor = a \qquad (b-a \geq a \geq 2)$$

$$\left\lceil \frac{a-1+x}{2} \right\rceil \leq \left\lceil \frac{a-1+b}{2} \right\rceil \leq \left\lceil \frac{2b-3}{2} \right\rceil = b-1$$

So SHARES do not add new coins.

Same with ROOT OPS (add or remove many root).

Updates add at most 2 coins.



□

[Only updates add coins. They add at most 2 coins each.
Each SPLIT and FUSE takes away at least one coin.
Hence # splits + # fuses ≤ 2 · # updates]