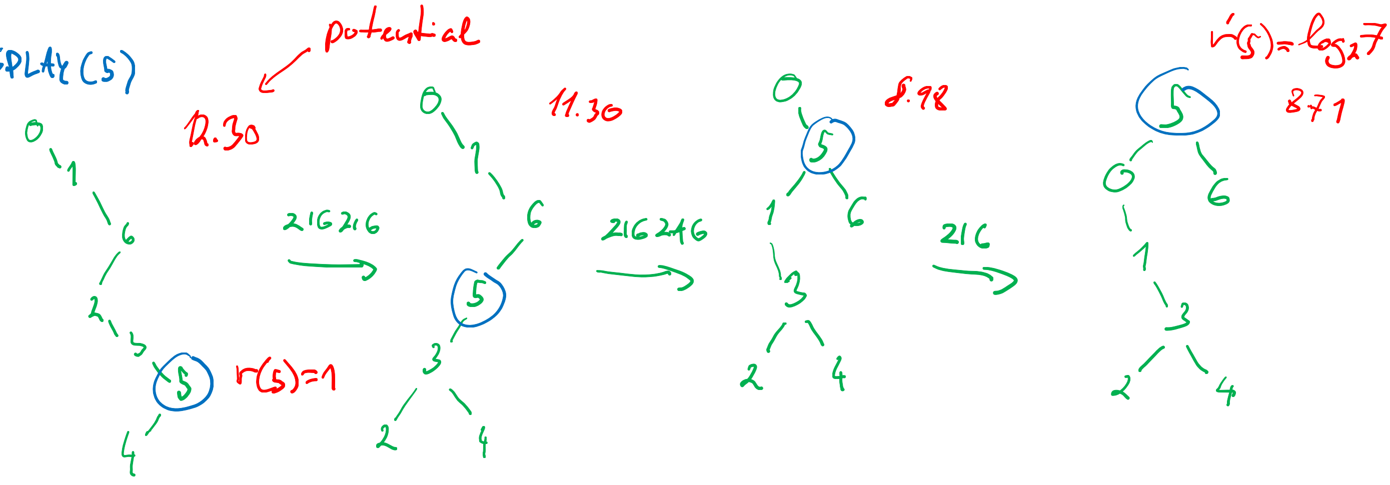
Splay trees (cont.)

Ex: SPLAY(5)



Notation: v ... node, n ... # nodes, $T(v)$ ... subtree rooted at v

$s(v)$ ... # nodes in $T(v)$ <u>size</u> $1 \le s(v) \le n$

$r(v) := \log_2 s(v)$ <u>rank</u> $0 \le r(v) \le \log n$

$\Phi := \sum_v r(v)$ <u>potential</u> $0 \le \Phi \le n \log n$

real cost of SPLAY = # rotations (1 for zig, 2 for zigzig, 2 for zigzag)

unit cost

Thm: The amortized cost of SPLAY(x) is at most $3(r'(x) - r(x)) + 1 = O(\log n)$ where $r(x)$ and $r'(x)$ are the ranks of x before and after.

(proof postponed)

Cor: A sequence of m SPLAYs runs in $O((n+m)\log n)$ time.

Pf: $\sum_{i=1}^{m} R_i \le \sum_{i=1}^{m} A_i - \Delta\Phi$, $\Delta\Phi \le n \log n$ for the potential drop $\square$

operations in splay trees

<u>idea</u>: always splay the lowest visited node

- FIND(x): search(x), SPLAY the node where we stopped (x if FIND successful)

real cost: $O(d)$ $O(d)$ d = depth of

amortized: $O(\log n) \Leftarrow O(\log n)$ (by Thm)

(search accounted to SPLAY, potential rescaled by some const.)

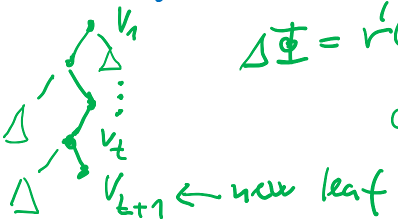
$\Rightarrow$ FIND has $O(\log n)$ amortized time.

- INSERT(x) : search(x), add a leaf x, SPLAY(x)
  (if search unsuccessful)

  similar to FIND up to: $O(1)$ real cost but increases potential!

  Lemma: Adding a leaf increases potential by $O(\log n)$.

  Pf.



$$\Delta\Phi = \underbrace{r'(v_{t+1})}_{=0} + \sum_{i=1}^{t} (r'(v_i) - r(v_i)) \le r'(v_1) - r(v_t) = O(\log n)$$

↑ telescoping

$$r'(v_i) = \log(s(v_i)+1) \le \log s(v_{i-1}) = r(v_{i-1}) \quad \square$$

$\Rightarrow$ INSERT has $O(\log n)$ amortized time.

- DELETE(x): search(x), remove a leaf x / internal x with 1 child / min in the right subtree of x,
  SPLAY(y)
  ↑
  (the parent of the removed)

Again, the real cost of search(x) or search(x) + succ(x) can be accounted to SPLAY(y). The 2nd step only decreases potential.
($\Delta\Phi < 0$)

$\Rightarrow$ DELETE has $O(\log n)$ amortized time.

Thm: A sequence of m operations FIND, INSERT, DELETE on initially empty splay tree takes time $O(m \log n)$ where n is the max. number of nodes during the sequence.

Pf. $$\sum R_i = \sum A_i - \Delta\Phi = O(m \log n) - \underbrace{\Phi_m}_{\ge 0} + \underbrace{\Phi_0}_{=0} \quad \square$$

Analysis of SPLAY

Thm: The amortized cost of SPLAY(x) is at most $3(r'(x) - r(x)) + 1 = O(\log n)$ where $r(x)$ and $r'(x)$ are the ranks of x before and after.

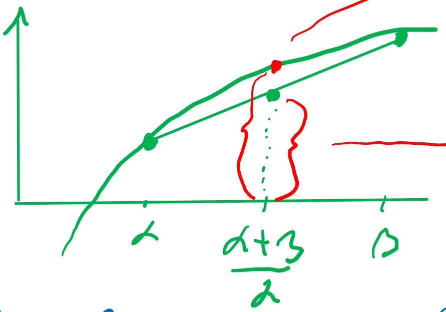
telescope

Pf: Claim: ZIGZAG(x) / ZIGZIG(x) has amortized cost $\le 3(r'(x) - r(x))$
ZIG(x) —"— $\le 3(r'(x) - r(x)) + 1$.

Lemma: For any $\alpha, \beta > 0$, $\log \frac{\alpha+\beta}{2} \geq \frac{\log \alpha + \log \beta}{2}$.

$\Uparrow$ concavity of log

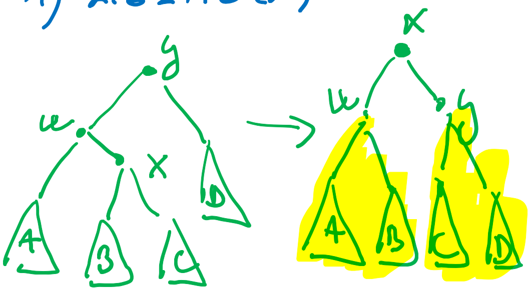
Pf:



[Jensen's inequality special case] □

$$\Longrightarrow \log \alpha + \log \beta \leq 2 \log(\alpha+\beta) - 2$$ ← to compensate real cost (*)

1) zigzag(x)



real cost ↘

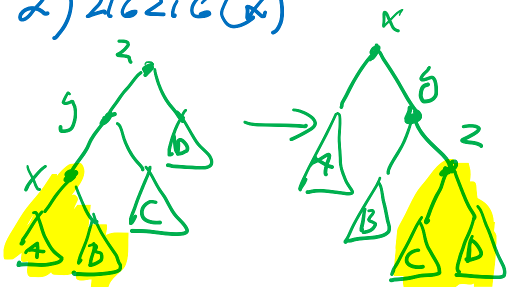
$$A = 2 + r'(w) + r'(x) + r'(y) - r(w) - r(x) - r(y)$$

$r'(w) + r'(y) \leq 2r'(x) - 2$ by (*)

$r(w) + r(y) \geq 2r(x)$ by $T(x) \subseteq T(w), T(y)$ (monotonicity)

$$\Longrightarrow A \leq 3(r'(x) - r(x))$$ ✓

2) zigzig(x)



real cost ↘

$$A = 2 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z)$$

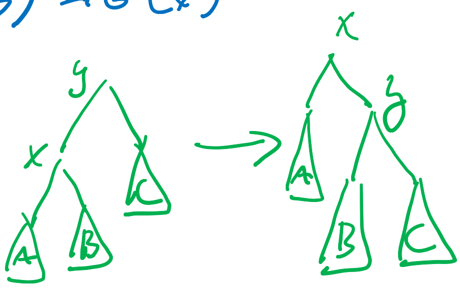
$r(x) + r'(z) \leq 2r'(x) - 2$ by (*)

$r'(y) \leq r'(x)$, $r(y) \geq r(x)$, $r(z) = r'(x)$

by $T'(y) \subseteq T'(x)$, $T(y) \supseteq T(x)$, $s(z) = s'(x)$

$$\Longrightarrow A \leq 3(r'(x) - r(x))$$ ✓

3) zig(x)



real cost

$$A = 1 + r'(x) + r'(y) - r(x) - r(y)$$

$r'(y) \leq r'(x)$, $r(y) \geq r(x)$

by $T'(y) \subseteq T'(x)$, $T(y) \supseteq T(x)$

(only for telescoping)

$$\Longrightarrow A \leq 2(r'(x) - r(x)) + 1 \leq 3(r'(x) - r(x)) + 1$$

by $r'(x) \geq r(x)$ ✓

□

Weighted analysis (to compare with optimal static trees)

generalized potential: $w(x) > 0$ weight of x

$s(x) = \sum_{y \in T(x)} w(y)$ size

$r(x) = \log_2 s(x)$ rank

$\Phi = \sum_x r(x)$ potential

(the same)

Lemma: The (weighted) amortized cost of SPLAY(x) is at most (total weight)
$3(r'(x) - r(x)) + 1 = O(\log(W/w(x)) + 1)$ where $W = \sum_x w(x)$.

Pf: Since all weights > 0, monotonicity holds $\Rightarrow$ the previous proof works.
$r'(x) = \log W$, $r(x) = \log s(x) \leq \log w(x)$ $\square$

Optimal static tree

T ... BST, $c_T(x)$ = # nodes visited when accessing x in T (depth + 1)

$p(x)$ ... probability of accessing x (distribution)

average access cost in T = $\mathbb{E}_x(c_T(x)) = \sum_x p(x) c_T(x)$ (expectation)

OPT (optimal static tree) = the tree that minimizes

Note: given p, OPT for $x_1, \dots, x_n$ can be constructed in $O(n^3)$ time by a dynamic alg. ($O(n^2)$ using Knuth's inequality)

Splay tree vs OPT (static optimality)

set $w(x) := p(x)$, $W = 1$

$\Rightarrow$ amortized cost of access to x = $O(\log(1/p(x)) + 1)$

(by Lemma)

$\Rightarrow$ expected amortized cost $\mathbb{E}[O(\log 1/p(x) + 1)] = O(1 + \underbrace{\sum_x p(x) \log 1/p(x)}_{H(p) \text{ entropy}})$

However, it can be shown (by expected length of a ternary prefix-free code) that $\mathbb{E}(c_{OPT}(x)) \geq H_3(p) = \Theta(H(p))$

ternary entropy

$\Rightarrow$ SPLAY tree is worse than OPT only by a multiplicative constant! (not knowing anything about p)