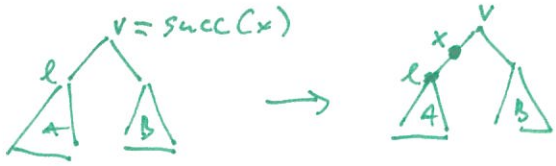
Alternative INSERT + DELETE



only $r'(v)$, $r'(x)$ are modified

$\Rightarrow \Delta\Phi = O(\log n)$

• INSERT(x): finds $v = \text{succ}(x)$ / pred(x), SPLAY(v), add x

$\Rightarrow O(\log n)$ amortized cost

• DELETE(x): FIND(x), SPLAY(x), remove x, $y = $ MIN(B), SPLAY(y)

FIND(x) SPLAY(x)

$O(\log n)$ amortized

$\Phi(L) + \Phi(R) \leq \Phi$

decreases Φ

$\Delta\Phi \leq 0$

$y = $ MIN(R)

SPLAY(y)

$O(\log n)$ amortized

$\Delta\Phi = r'(y) - r(y) = O(\log n)$

$\Rightarrow O(\log n)$ amortized cost

Analysis of SPLAY

Theorem: The amortized cost of SPLAY(x) is at most $3(r'(x) - r(x)) + 1$.

Claim: ZIG ZAG / ZIG ZIG amortized cost $\leq 3(r'(x) - r(x))$

ZIG — $\leq 3(r'(x) - r(x)) + 1$

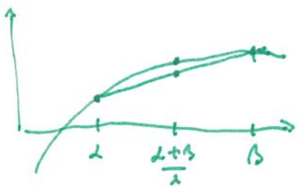
(telescoping)

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

$\Uparrow$ concavity of log (second derivative is $-1/x^2$, negative)

[Jensen's inequality special case]

Pf:



$\Rightarrow \log \alpha + \log \beta \leq 2 \log(\alpha+\beta) - 2$ (1) → to compensate the real cost

ZIG ZAG:

$$A \leq \underbrace{2}_{\text{(real cost)}} + \underbrace{r'(w) + r'(x) + r'(y) - r(w) - r(x) - r(y)}_{(\Delta\Phi)}$$

$$r'(w) + r'(y) = \log s'(w) + \log s'(y) \overset{(1)}{\leq} 2 \log(\underbrace{s'(w) + s'(y)}_{\leq s'(x)}) - 2 \leq 2r'(x) - 2$$

$$r(w) + r(y) \geq 2r(x) \quad \text{since } T(x) \subseteq T(w), T(y)$$

$$\Rightarrow A \leq 2 + 2r'(x) - 2 + r'(x) - r(x) - 2r(x) = 3(r'(x) - r(x)). \checkmark$$

ZIG ZIG:

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

all vertices b/f y $\leftarrow$ $r(x) + r'(z) = \log s(x) + \log s'(z) \le 2\log(\underbrace{s(x)+s'(z)}_{\le s'(x)}) - 2$

$$\le 2r'(x) - 2$$

$$\Rightarrow A \le 3r'(x) + \underbrace{r'(y)}_{\le r'(y)} - 2r(x) - \underbrace{r(y)}_{\ge r(x)} - \underbrace{r(z)}_{= r'(x)} \le 3(r'(x) - r(x)). \checkmark$$

ZIG:

$$A = 1 + r'(x) + \underbrace{r'(y)}_{\le r'(x)} - r(x) - \underbrace{r(y)}_{\ge r(x)} \le 1 + 2r'(x) - 2r(x) \le 3(r'(x) - r(x)) + 1$$

as $r'(x) - r(x) \ge 0$ $\quad T(x) \subseteq T'(x)$

$\square$

<u>Weighted analysis</u> $\leftarrow$ to compare with static optimal trees.

generalized potential:

$w(x) > 0$ weight of x (ex: ~~$w(x) = p(x)$~~) ... prob. distribution

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

can be arbitrarily small / large $\rightarrow$ $r(x) = \log_2 s(x)$ rank

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

<u>Lemma</u>: The (weighted) amortized cost of SPLAY(x) is at most

$$3(r'(x) - r(x)) + 1 = O(\log(W/w(x)) + 1) \quad \text{where } W := \sum_x w(x).$$

Pf: Since all weights positive, $T(u) \subseteq T(v) \Rightarrow s(u) \le s(v)$, $r(u) \le r(v)$. (monotonicity)

$\Rightarrow$ the previous proof generalizes.

$r'(x) = \log W$, $r(x) = \log s(x) \le \log w(x)$ implies $\square$

<u>Ex.</u> $w(x) = 1/n$ (uniform distribution) $\Rightarrow W = 1$

$1/n \le s(x) \le 1$, $-\log n \le r(x) \le 0$, $-n\log n \le \Phi \le 0$.

$\Rightarrow$ SPLAY(x) amortized $O(\log 1/(1/n) + 1) = O(\log n)$

$\Rightarrow$ sequence m splays real cost $O((m+n)\log n)$.

$n \log n$ for $-\Delta\Phi$.

Optimal static BST (intermezzo)

NO (INSERT / DELETE) / updates

T … BST for $\{x_1, \dots x_n\}$, $c_T(x)$ = # nodes visited when accessing x in T ($c_T(x) = 1$ if x root)

P … prob. distribution, $p(x)$ … prob. of accessing x

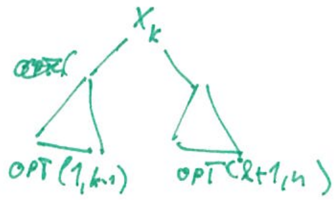
average access cost in T = $\mathbb{E}_x(c_T(x))$ … minimized by OPT <u>optimal static BST</u>

Construction of OPT = OPT(1, n)

OPT(i, j) … optimal for $\{x_i, \dots, x_j\}$ (empty if $i > j$)

c_{ij} = "$\mathbb{E}_x(c_{OPT(i,j)}(x))$" optimal cost

$r_{ij} = k$ s.t. x_k is the root of OPT(i, j)



dynamic algorithm: $\Rightarrow O(n^3)$ time

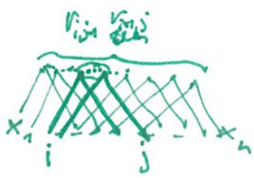
$c_{ii} := p(x_i)$, $r_{ii} = i$

for $l = 1 \dots n-1$ // interval length − 1

for $i = 1 \dots n-l$, $j = i+l$ // int. start, end

$$c_{ij} = \min_{i \le k \le j} c_{i,k-1} + c_{k+1,j} + \sum_{i \le m \le j} p(x_m)$$

$$r_{ij} = \operatorname{arg\,min}_{i \le k \le j} \text{ —"— }$$



Knuth's inequality: $r_{i,j-1} \le r_{ij} \le r_{i+1,j} \Rightarrow O(n^2)$ time

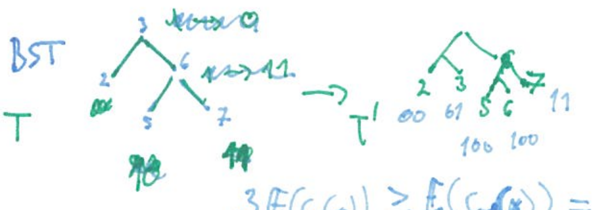
(exercise)

Splay trees vs optimal static BST (information theoretic bound)

$w(x) = p(x)$ … given prob. distribution p, $W = 1$

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

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



prefix-free code

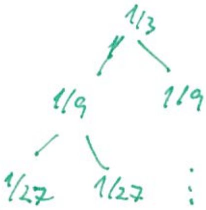
$3\mathbb{E}(c_T(x)) \ge \mathbb{E}(c_{T'}(x)) = \mathbb{E}(\ell(x)) \ge H(p) \Rightarrow = O(\mathbb{E}(c_{OPT}(x)))$

Theorem (static optimality): For any static BST T on a set X and any access sequence, where $f(x)>0$ is the frequency (# occurrences) of $x \in X$, the total cost of accesses in a Splay tree is $O\left(\sum_x f(x)\, c_T(x)\right)$.

$c_T(x)$: total cost in T

Pf: We set $w(x) := 3^{-c_T(x)}$ $\Rightarrow W = \sum_x w(x) < \sum_{i=1}^{\infty} \frac{2^{i-1}}{3^i} = \frac{1}{2}\sum_{i=1}^{\infty}\left(\frac{2}{3}\right)^i = 1$

infinite tree ($\sum_{i=1}^{\infty} q^i = \frac{q}{1-q}$ for $0 \le q < 1$)



$\Rightarrow r(x) = \log s(x) \le \log W < 0$

$\Phi = \sum_x r(x) < 0$

$\Rightarrow r(x) = \log s(x) \ge \log w(x) = \Theta(-c_T(x))$

$\Phi = \sum_x r(x) \ge -\Theta\left(\sum_x c_T(x)\right)$

By Lemma, amortized cost SPLAY(x) is

$$O(\log(W/w(x)) + 1) = O(\log 3^{c_T(x)} + 1) = O(c_T(x)+1) = O(c_T(x))$$

So, total real cost of access in SPLAY(x) is

$$O\left(\sum_x f(x)\, c_T(x)\right) - \underbrace{\Delta\Phi}_{\Phi_m - \Phi_0} = O\left(\sum_x f(x)\, c_T(x)\right) \underbrace{-\Phi_m}_{O(\sum c_T(x))} + \underbrace{\Phi_0}_{<0} = O\left(\sum_x f(x)\, c_T(x)\right)$$

as $f(x) > 0$ ∀

□

Open Question: Are splay trees also dynamically optimal?
(within a constant factor of any dynamic tree that knows the sequence ~~ahead~~ beforehand)

Working set bound

idea: accessed items drop since last time?

access sequence $x_1 \dots x_m$ — distinct elements accessed since last occurrence of x_i (3, $x_i := 3$)

$z_i = |$working set of $x_i|$ = distinct elements,
= all previous items if 1st occurrence

Theorem (working set bound): A splay tree on n items processes any access sequence $x_1 \dots x_m$ in time $O\left(n \log n + m + \sum_{i=1}^{m} \log(1+z_i)\right)$.

(no proof) $\Rightarrow$ good for caches

~~idea: $w(x) = \frac{1}{z_i}$~~

[**Note**: Splay trees good for nonuniform distributions, other trees (AVL ...) win on constants on uniform distributions.