Advanced Data Structures

Lecture 7

Lecture by Dr. Shay Mozes
Typeset by Steven Karas

2016-12-22
Last edited 20:49:06 2016-12-22

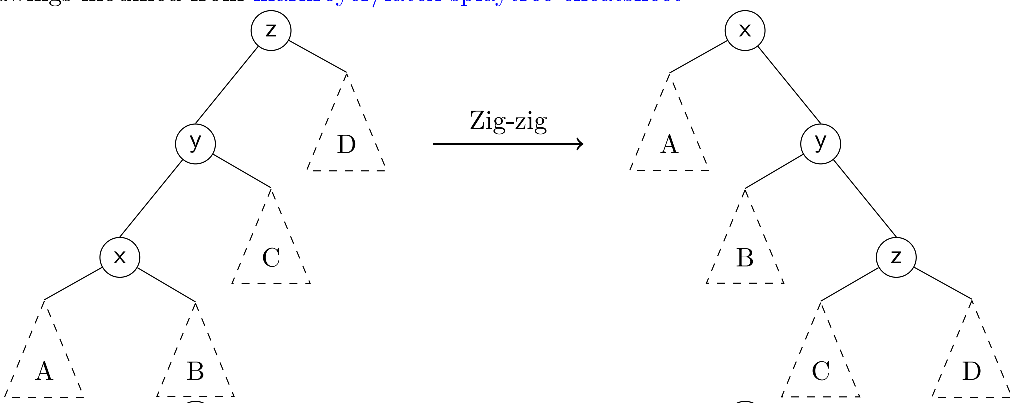
Errata: Move to Root O(1)-competitive with stochastic model. Not competitive with static model

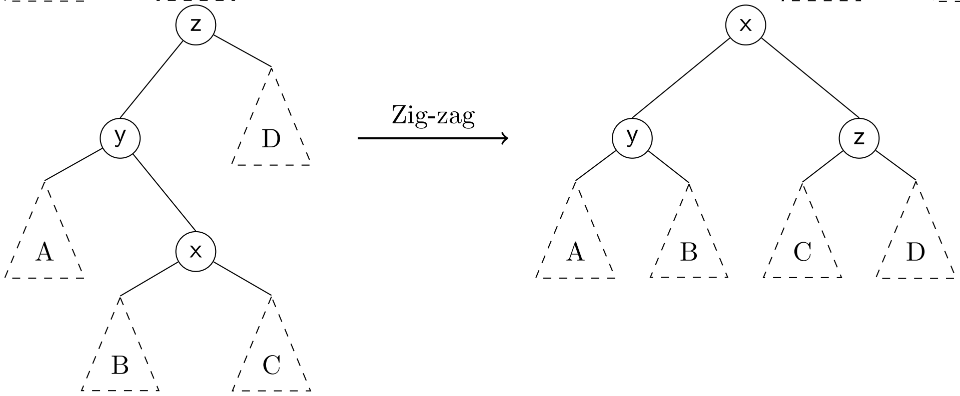
1 Splay Trees

Introduced in Sleator, Tarjan 1985

1.1 Primitives

Drawings modified from markroyer/latex-splaytree-cheatsheet





1.2 Concept

When we search for x, we move it up to the root using the above primitives (any maybe a single rotation). We will show that the amortized access time is $O(\log n)$.

1.3 Potential Function

Let $S(x)$ = size of the subtree rooted at x. Let $r(x) = \log(S(x))$.

$$\Phi(T) = \sum_{x \in T} r(x)$$

Potential of a full binary tree $r(x_i) = \frac{n}{2^i} \cdot \log(2^{i-1})$ for each node x_i at depth i where i is 0 at the root.

$$\Phi(T) = \sum_{i=1}^{\log n} \frac{n}{2^i} \cdot \log(2^{i-1}) = n \sum \frac{i-1}{2^i} = \Theta(n)$$

Potential of a string

$$\Phi(T) = \sum_{i=1}^{n} \log(i) = \Theta(n \log n)$$

1.4 Amortized Access

Actual Cost We pay the depth to find an element, and half the depth for zigzig and zigzag splaying.

$$\mathit{cost}(\text{zig-zig}) = 2 + \Delta\Phi$$

r changes for x, y, and z. $r'(x) = r(z)$, $r(y) \geq r(x)$, and $r'(x) \geq r'(y)$. Also, $S(x) + S'(z) < S'(x)$.

$$\begin{aligned}\mathit{cost}(\text{zig-zig}) &= 2 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z) \\ &\leq r'(x) + r'(z) - 2r(x) = (*)\end{aligned}$$

Note that the log function is convex, such that $\frac{\log a + \log b}{2} < \log(\frac{a+b}{2})$ for any a, b.

$$\begin{aligned}\frac{r(x) + r'(z)}{2} = \frac{\log S(x) + \log S'(z)}{2} &\leq \log\left(\frac{S(x) + S'(z)}{2}\right) \\ &\leq \log\frac{S'(x)}{2} = r'(x) - 1\end{aligned}$$

$$\begin{aligned}r'(z) &\leq 2r'(x) - 2 - r(x) \\ (*) &\leq 3r'(x) - 3r(x)\end{aligned}$$

$$\mathit{cost}(\text{zig-zag})^1 \leq 2(r'(x) - r(x)) \leq 3r'(x) - 3r(x)$$

[1]Not proven in class, and not a particularly interesting proof either...

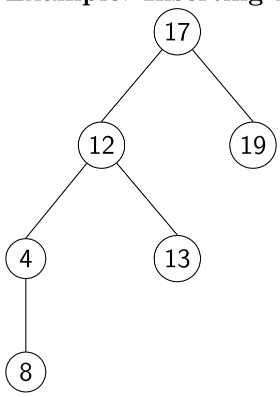
Cost of splay

$$\sum_{\substack{\text{zig-zig} \\ \text{zig-zag}}} cost(op) \leq \underbrace{\sum 3(r'(x) - r(x))}_{\text{series cancels out}} = 3(r(root) - r(x)) \leq 3\log n$$

1.5 Insertion

Insert x in its correct place, and then splay x.

Example: Inserting 13



1.6 Search

Look for x and splay the last element found while looking.

1.7 Join

Combine T_1 and T_2 where all the elements in T_1 are smaller than T_2. Find the largest element x in T_1 and splay it. Then connect T_2 as the right child of x

1.8 Split

Given an element x, find it in the tree, and splay it. Then return the children.

1.9 Delete

Search for x, splay it, and then return the join of the children.

1.10 Weighted analysis

Let $w(x)$ be some weight function for each node x in the tree. Instead of using $S(x)$ as the size of the subtree rooted at x, let $S(x) = \sum_{y \in T_x} w(y)$. Note that we've already shown the case where $w(x) = 1$ for every x, in which case the amortized cost is $O(\log n)$.

staticOPT Recall that for a sequence of operations σ where x appears f_x times:

$$staticOPT = \Theta\left(\sum \frac{f_i}{m} \log\left(\frac{m}{f_i}\right)\right)$$

$$\text{entropy} = \sum p_i \log \frac{1}{p_i}$$

Let $w(x) = f_x$.

$$amort(splay(x)) \leq 3(\underbrace{\log(m)}_{\sum f_y = m} - \log(S(x))) \leq 3(\log m - \log f_x)$$

$$= O\left(\log \frac{m}{f_x}\right)$$

$$amort(\sigma) = \frac{1}{m} \sum f_x \log \frac{m}{f_x} = \sum \frac{f_x}{m} \log\left(\frac{m}{f_x}\right)$$

Static Finger Theorem Let $dist(x, y) =$ the number of elements between x and y (inclusive) on a path through the tree. For any constant element f the amortized operation cost is $\log(dist(x, f))$. In other words, for any element x the amortized cost is

Proof Let $w(x) = \frac{1}{dist(x,f)^2}$[2].

$$S(root) \leq 2 \sum_{i=0}^{\infty} \frac{1}{i^2} = O(1)$$

$$S(x) \geq w(x) \geq \frac{1}{dist(f, x)^2}$$

$$r(x) \geq -2 \log n$$

Which gives us that the amortized cost is $O(\log dist(x, f))$

Working Set Theorem The amortized access time to an element x is $O(\log D)$ where D is the number of distinct elements that have been accessed since the last access of x.[3]

Sequential Access Theorem The amortized cost to access each element in order is $O(n)$ without regard to the initial state of the tree.

Dynamic Finger Theorem Immediately following $splay(y)$ the amortized cost of $splay(x)$ is $O(\log dist(y, x))$.

dynamicOPT **Conjecture** Splay trees are $O(1)$-competitive with $dynamicOPT$.

[2] Even Shay admitted he has no idea where in the hell this came from, and that the potential function can become negative, but that it is bound, so it's not that bad

[3] Unproven in the lecture because the weight function is dynamic