**15-451/651: Design & Analysis of Algorithms**
**Some more notes on Splay Trees**

Here's a more verbose statement of the Access Lemma, and its use to prove the Balance Theorem.

## 1 Access Lemma

Take any tree $T$ (it does not have to arise in the splaying algorithm, and may be created any which way), and any weights $w(\cdot)$ on the nodes of $T$. Let $T(x)$ be the subtree rooted at $x$.

For a node $x \in T$, define

$$s(x) := \sum_{y \in T(x)} w(y)$$
$$r(x) := \lfloor \log_2 s(x) \rfloor$$
$$\Phi(T) := \sum_{x \in T} r(x)$$

**Lemma 1 (Access Lemma)** *Take any tree $T$ with root $t$, and any weights $w(\cdot)$ on the nodes. Suppose you splay node $x$; let $T'$ be the new tree. Then*

$$\begin{aligned}\textit{amortized number of splaying steps } &= \textit{actual number of splaying steps } + (\Phi(T') - \Phi(T)) \\ &\leq 3(r(t) - r(x)) + 1.\end{aligned}$$

To get a sense for the Access Lemma, let us use it to prove the Balance Theorem.

## 2 Balance Theorem

Suppose all the weights equal 1. Then the access lemma says that if we splay node $x$ in tree $T$ to get the new tree $T'$

$$\begin{aligned}\text{actual number of splaying steps } + \big(\Phi(T') - \Phi(T)\big) &\leq 3(\log_2 n - \log_2 |T(x)|) + 1 \\ &\leq 3 \log_2 n + 1.\end{aligned}$$

Hence, if we start off with a tree $T_0$ of size at most $n$ and perform any sequence of $m$ splays to it

$$T_0 \xrightarrow{splay} T_1 \xrightarrow{splay} T_2 \xrightarrow{splay} \cdots \xrightarrow{splay} T_m,$$

repeatedly using this inequality $m$ times shows:

$$\text{actual total number of splaying steps } + (\Phi(T_m) - \Phi(T_0)) \leq m(3 \log_2 n + 1).$$

In any tree $T$ with unit weights, each $s(x) \leq n$ so each $r(x) \leq \log_2 n$ so $\Phi(T) \leq n \log_2 n$; also $\Phi(T) \geq 0$. Rearranging, we get

$$\begin{aligned}\text{actual total number of splaying steps } &\leq m(3 \log_2 n + 1) + (\Phi(T_0) - \Phi(T_m)) \\ &\leq O(m \log n) + O(n \log n).\end{aligned}$$

This proves the **Balance Theorem**.

## 3 What next?

Suppose you have some way of inserting a node into a tree. Now if you start off with a tree $T_0$ of size $n_0$ and perform any sequence of $m$ splays and inserts to it

$$T_0 \xrightarrow{splay} T_1 \xrightarrow{splay} T_2 \xrightarrow{insert} T_3 \xrightarrow{splay} T_4 \cdots \xrightarrow{insert} T_m,$$

you now know how the potential changes during the splay moves. How does it change for the insert moves? You'll solve this in HW#4. (What if you had deletes? Think about it if you're interested.)