Addendum for Data Structures and Algorithm Analysis Thm 11.5

October 17, 2010

The proof of the splaying bound describes an $O(\log N)$ amortized bound per splay step. Although this is most intricate part of the splaying analysis, to show that insertions and deletions are $O(\log N)$, potential changes that occur either prior to or after the splaying step should be accounted for.

In the case of insertion, assume we are inserting into an $N-1$ node tree. Thus, after the insertion, we have an N-node tree, and the splaying bound applies. However, the insertion at the leaf node adds potential prior to the splay to each node on the path from the leaf node to the root. Let $n_1, n_2, ..., n_k$ be the nodes on the path prior to the insertion of the leaf (n_k is the root), and assume they have sizes $s_1, s_2, ..., s_k$. After the insertions, the sizes are $s_1+1, s_2+1, ..., s_k+1$. (The leaf will contribute 0 to the potential so we can ignore it). Note that (excluding the root node) $s_j + 1 \leq s_{j+1}$, so the new rank of n_j is no more than the old rank of n_{j+1}. Thus, the increase of ranks, which is the maximum increase in potential that results from adding a new leaf, is limited by the new rank of the root, which is $O(\log N)$.

A deletion consists of a non-splaying step that attaches one tree to another. This does increase the rank of one node, but that is limited by $\log N$ (and is compensated by the removal of a node, which at the time was a root). Thus the splaying costs accurately bound the cost of a deletion.