Illinois Institute of Technology
Department of Computer Science

# Homework Assignment 4

CS 535 Design and Analysis of Algorithms
Fall Semester, 2016

**Due: Thursday, September 22, 2016**

**Remember the Honesty Pledge!**

1. We could get the same effect as splaying (moving the accessed item to the root) using only zig and zag steps, not bothering with the more complicated zig-zig, zig-zag, zag-zig, and zag-zag steps. If we do that, we only need Case 1 on pages 4–5 in the notes. What is the amortized cost of an access with such simple splaying?

2. Consider a new operation on splay trees, `cut`$(i_1, i_2, T)$ that deletes all items $i$, $i_1 \leq i \leq i_2$ from the splay tree $T$. Implement the `cut` operation so that the amortized cost of a `cut` is logarithmic and analyze the amortized cost as in the table at the bottom of page 10 in the notes. (*Hint*: This problem is easy.)

3. In this exercise we prove that a splay tree is as good, to within a constant multiple, of any static binary search tree.

   (a) Given any binary search tree, let $d(x)$ be the depth of (an internal) node $x$, that is, the number of internal nodes on the path from the root to $x$, so $d(\text{root}) = 1$ and the cost of accessing $x$ in the given tree is $O(d(x))$. Prove the pseudo-Kraft inequality $\sum_{\text{internal nodes } x \in T} 3^{-d(x)} \leq 1$ for any finite binary tree. Prove that the sum is exactly 1 for the complete infinite binary tree.

   (b) Define the weight of a node $x$ in a splay tree as $w(x) = 3^{-d(x)}$. Prove that the amortized cost of accessing item $x$ in a splay tree is $O(1 + d(x))$.

4. Derive the bound for `insert` given in the table at the top of page 11 of the splay tree notes.

5. **PhD Qualifying Exam Section Problem 4.**

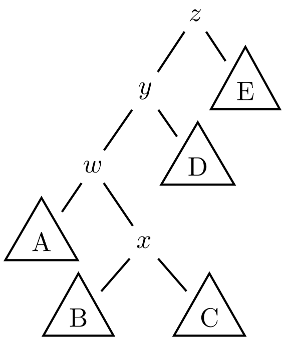
   Redefine the potential of a splay tree to be

   $$\Phi(T) = \sum_{\text{nodes } x \in T} \alpha r(x),$$

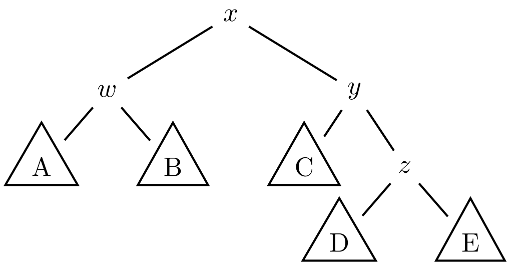
   where $\alpha = 3/\lg(27/4) \approx 1.089$.

   (a) Using this new potential function, prove that the amortized cost of the `ZIG-ZIG` step at $x$ is at most $3\alpha[r'(x) - r(x)]$. (*Hint*: In the notes we proved that $2 + r'(x) + r'(y) + r'(z) - r(x) - r(y) - r(z) \leq 3[r'(x) - r(x)]$; multiply both sides of this inequality by $\alpha$.)

   (b) Consider the more complicated rule, the `ZAG-ZIG-ZIG` step which uses 3 rotations (at $w$, then at $z$, then at $y$) to transform



into



We want to derive an upper bound of $3\alpha[r'(x) - r(x)]$ on the amortized cost of such a step.

i. Prove that the amortized cost of the `ZAG-ZIG-ZIG` step is

$$\hat{c} = 3 + \alpha[r'(x) + r'(y) + r'(z) + r'(w) - r(x) - r(y) - r(z) - r(w)]$$

and that we can get the desired bound on the amortized cost of such a step by proving that

$$\alpha[3r'(x) - 2r(x) - r'(y) + r(y) - r'(z) - r'(w) + r(w)] \geq 3.$$

ii. Using the fact that $r(y) \geq r(w) \geq r(x)$, show that this last inequality follows from proving that

$$3r'(x) - r'(y) - r'(z) - r'(w) \geq 3/\alpha,$$

which follows in turn from

$$-2\lg\frac{S'(y)}{S'(x)} - \lg\frac{S'(w)}{S'(x)} \geq 3/\alpha.$$

Explain why this inequality must hold.