ICS 261—Sample problems on LCAs and augmented tree structures

David Eppstein, ICS, UC Irvine

February 25, 1999

1. Suppose that instead of least common ancestor queries in a tree, we only need to answer ancestor-descendant queries: is node x an ancestor of node y? (For instance, this sort of query is what one would need to route packets on a tree-shaped computer network.) Describe a simple data structure for answering these queries in constant time.

   Hint: this is much easier than the LCA algorithm described in class.

2. Suppose we wish to maintain a tree, subject to cutting and linking operations, and answer LCA queries: given two vertices as input, output their lowest common ancestor in the current tree. Describe how to modify Tarjan's cutting and linking trees data structure to solve this problem in $O(\log n)$ amortized time per update or query.

   Hint: you don't need to store any auxiliary information in the splay tree nodes; the only reason you need the splay trees at all is to keep track of which edges belong to which paths.

3. Recall that the *standard deviation* of a set of n numbers $x_1, x_2, \ldots x_n$ is defined as

$$\sqrt{\frac{\sum x_i^2 - (\sum x_i)^2/n}{n-1}}.$$

   (a) Show how to preprocess an array of n items $A[0], \ldots A[n-1]$ in linear time, so that one can compute the standard deviation of any contiguous subarray in constant time. *Hint: this is just arithmetic, no complicated data structures are needed. You can assume that square roots can be calculated in constant time.*

   (b) Describe how to maintain an array of n items, subject to updates that change the value of a single item, and answer queries that request the standard deviation of a contigous subarray, in time $O(\log n)$ per update or query.