

## 6.897 ADVANCED DATA STRUCTURES (SPRING'05)

Prof. Erik Demaine      TA: Mihai Pătrașcu

Problem 2      *Due: Monday, Feb. 14*

**Preliminaries.** Remember the definition of a rotation from Lecture 3. In this problem, we will work with an alternative cost model for BST algorithms. Roughly, the cost is just the number of rotations performed. More precisely, a query must bring the queried element to the root; the cost of the query is 1 plus the number of rotations performed. Rotations may be performed anywhere in the tree, for the same cost of 1. We are only interested in this BST cost metric, and you do not need to worry about the actual running times of your algorithms. (If you are curious, all bounds in this model also hold for the model defined in class, up to constant factors. Therefore, one generally uses whichever model is easier to analyze for a certain problem.)

**Prove the following:** Suppose we are given two BSTs  $T_1$  and  $T_2$  on the same  $n$  keys. Show that one can transform  $T_1$  into  $T_2$  using at most  $2n + O(1)$  rotations.

*Hint:* There is a simple, short, and non-messy solution to this question—and probably a lot of messy ones. Try to find the simple one.

**Competitiveness with an  $O(\lg n)$  guarantee.** Assume that we perform  $m \geq n$  operations. Suppose that we have an  $\alpha$ -competitive BST algorithm, i.e., the cost (number of rotations) is at most  $\alpha \cdot \text{OPT}$ , where  $\text{OPT}$  is the optimal cost for the given access sequence. Note that  $\alpha$  may not be a constant (think of  $\alpha = O(\lg \lg n)$ , as in Lecture 4). Therefore, if  $\text{OPT}$  is big (close to  $m \lg n$ ),  $\alpha \cdot \text{OPT}$  may even be higher than the trivial  $O(m \lg n)$  guarantee. We would like to have the best of both worlds: keep the  $\alpha$ -competitiveness, but always guarantee an  $O(\lg n)$  amortized running time.

**Prove the following:** Given an online  $\alpha$ -competitive BST algorithm, one can construct an online BST algorithm whose cost is always  $O(\min\{\alpha \cdot \text{OPT}, m \lg n\})$ .

*Hint:* Break the access sequence into chunks of  $n$  accesses. Remember that your BST algorithm must be online, so it doesn't know the access sequence ahead of time. Not even  $m$  is known ahead of time! (But it is guaranteed that  $m \geq n$ .)