

CSCI2100: Regular Exercise Set 10

Prepared by Yufei Tao

Problem 1*. Prove that an insertion into the AVL-tree can trigger at most one (single/double) rotation.

Problem 2**. Prove that it suffices to handle only 2-level imbalance in the insertion and deletion algorithms of the AVL-tree. In other words, neither algorithm will run into a situation where an imbalanced node sees an absolute difference of 3 or higher in the heights of its left and right subtrees.

Problem 3. Let T be a balanced binary tree of n nodes. For each node u of T , define its *count* as the number of nodes in its subtree (remember that the subtree includes the node itself). Modify the insertion and deletion algorithms to maintain the counts of all the nodes. Your algorithms must still perform an insertion and deletion in $O(\log n)$ time.

Problem 4. In this exercise, we will design an algorithm to detect whether network packets have been sent out in a wrong order. A network packet here is defined as a pair (t, k) where t is the timestamp when the packet was sent out, and k is an integer representing the packet's contents. These packets arrive in an arbitrary order (the order is not necessarily in ascending t , because the packets may have been transmitted at different speeds). Design an algorithm to detect whether you have received any two pairs (t_1, k_1) and (t_2, k_2) such that $t_1 < t_2$ but $k_1 > k_2$. You may assume that all the packets have distinct t -values and distinct k -values. Your algorithm must process every incoming packet in $O(\log n)$ time, where n is the number of packets received.

Problem 5**. In two-dimensional space, a point (x, y) *dominates* another point (x', y') if $x > x'$ and $y > y'$. Let S be a set of n points in two-dimensional space, such that no two points share the same x - or y -coordinate. A point $p \in S$ is a *maximal point* of S if no point in S dominates p . For example, suppose that $S = \{(1, 1), (5, 2), (3, 5)\}$; then S has two maximal points: $(5, 2)$ and $(3, 5)$.

Describe a data structure to support the following operations on a dynamic set S :

- INSERT(p): Adds a new point p to S .
- QUERY: Reports all the maximal points of S .

If n is the current size of S , your structure must support an insertion in $O(\log n)$ amortized time, and a query in $O(1 + k)$ time, where k is the number of maximal points.