Analysis of Algorithms I: Problem Assignment 3
Due on Gradescope at 11:59pm on Monday, March 7, 2022.

Instructions

- Problems 1-5 are each worth 10 points.
- Submit your solutions in pdf format. Late submissions will **not** be accepted.
- You can discuss with TAs or other students but you must acknowledge them at the beginning of each problem and your solutions must be written in your own words.

Problems

1. Prove the following useful property of a binary search tree (with distinct keys):

   Property 1. *Let x be a node in a BST T. Let* max *and* min *denote the largest and smallest keys in the subtree rooted at x, respectively. For any node y outside the subtree rooted at x, show that either $y.key >$* max *or $y.key <$* min*. This implies that if there is a key k in the tree that satisfies* $\min \leq k \leq \max$ *then it must lie inside the subtree rooted at x. (Here the subtree rooted at x includes x itself.)*

   Use it to solve Exercise 12.2-5, 12.2-6 and 12.2-9 on page 293. In all three exercises, we assume the BST has distinct keys.

2. Problem 13-2: Join operation. For a): you only need to answer the following two questions: 1) Let T be a red-black tree in which the root has black height T.bh. Then after an insertion, T.bh either stays the same or increases by 1. Describe the scenario when it increases by 1. 2) If a node z has black height h, use $O(1)$ time to compute the black height of z's children. Skip e) and f). Replace d) by the following: If $T_1.\text{bh} = T_2.\text{bh}$, what color should we make x to get a red-black tree? If $T_1.\text{bh} > T_2.\text{bh}$, what color should we make x so that properties $1, 2, 3$ and 5 are maintained? "Briefly" describe how to enforce property 4 in $O(\lg n)$ time.

3. Exercise 14.3-6 on page 354. You only need to describe the following key points: 1) What extra information to store in each node? 2) With this additional information in each node, how to answer Min-GAP efficiently? 3) Use Theorem 14.1 to prove that insertion and deletion of a node can still be done in $O(\lg n)$ time: Show that all the extra information for a node x can be derived from the information stored in its two children in $O(1)$ time.

4. Problem 16-1 (b) and (c) only on page 447.

5. Problem 16-2 on page 447.