CS120: Intro. to Algorithms and their Limitations Anshu & Hesterberg

Sender–Receiver Exercise 1: Reading for Senders

Harvard SEAS - Fall 2023 *2023-09-21*

The goals of this exercise are:

- to develop your skills at understanding, distilling, and communicating proofs and the conceptual ideas in them
- to practice reasoning about updates to dynamic data structures and binary search trees in particular

In the previous class (Tuesday 09-19), we saw that insert operations can be performed on a binary search tree (BST) in time $O(h)$, where h refers to the height of the tree. As an in-class exercise, some of you saw that a variety of different operations (search, min/max, next-smaller/next-bigger) can also be performed in time $O(h)$; pseudocode for those operations is in the detailed lecture notes. Here you will see how *deletions* can be performed in time $O(h)$.

Theorem 0.1. *Given a binary search tree T of height h, and a key K stored in the tree, we can delete a matching key-value pair (K, V) from T in time $O(h)$. Deletion means that we produce a new binary search tree that contains all of the key-value pairs in T except for one less occurrence of a pair with key K.*

For the proof, we will have you read Roughgarden II, Section 11.3.8 (attached), which has a particularly good description of the deletion operation. It's important to note a few small differences between Roughgarden's treatment of BSTs and ours:

- Roughgarden assumes that all of the keys are distinct; feel free to assume the same during this exercise.
- Roughgarden's Predecessor query is a bit different than our next-smaller query — it finds what's next-smaller than a key already in the tree, rather than what's next-smaller than an arbitrary query key K that's not necessarily in the tree.