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

We have already established most of Theorem 4.8 in the textbook, showing that a variety of different operations can be performed on BSTs in time $O(h)$, including `insert` updates. Here we will complete the proof, by showing that `delete` updates can also be done in time $O(h)$.

Theorem .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.*

To prepare for the exercise, we recommend reviewing Section 4.4 of the textbook to make sure you are comfortable with BSTs and how to implement the simpler operations on them (`insert`, `search`, `min/max`, and `predecessor/successor`).

The Proof