1. When node $z$ in Tree-Delete has two children, we could choose node $y$ as its predecessor rather than its successor. What other changes to Tree-Delete would be necessary if we did so? Some have argued that a fair strategy, giving equal priority to predecessor and successor, yields better empirical performance. How might Tree-Delete be changed to implement such a fair strategy?

2. Let us define a relaxed red-black tree as a binary search tree that satisfies red-black properties 1, 3, 4, and 5. In other words, the root may be either red or black. Consider a relaxed red-black tree $T$ whose root is red. If we color the root of $T$ black but make no other changes to $T$, is the resulting tree a red-black tree?

3. What is the largest possible number of internal nodes in a red-black tree with black-height $k$? What is the smallest possible number?

4. Argue that in every $n$-node binary search tree, there are exactly $n - 1$ possible rotations.

5. Show that any arbitrary $n$-node binary search tree can be transformed into any other arbitrary $n$-node binary search tree using $O(n)$ rotations. (Hint: First show that at most $n - 1$ right rotations suffice to transform the tree into a right-going chain.)

6. Consider a red-black tree formed by inserting $n$ nodes with RB-INSERT. Argue that if $n > 1$, the tree has at least one red node.