

17-4 The cost of restructuring red-black trees

There are four basic operations on red-black trees that perform **structural modifications**: node insertions, node deletions, rotations, and color changes. We have seen that RB-INSERT and RB-DELETE use only $O(1)$ rotations, node insertions, and node deletions to maintain the red-black properties, but they may make many more color changes.

a. Describe a legal red-black tree with n nodes such that calling RB-INSERT to add the $(n + 1)$ st node causes $\Omega(\lg n)$ color changes. Then describe a legal red-black tree with n nodes for which calling RB-DELETE on a particular node causes $\Omega(\lg n)$ color changes.

Although the worst-case number of color changes per operation can be logarithmic, we shall prove that any sequence of m RB-INSERT and RB-DELETE operations on an initially empty red-black tree causes $O(m)$ structural modifications in the worst case. Note that we count each color change as a structural modification.

b. Some of the cases handled by the main loop of the code of both RB-INSERT-FIXUP and RB-DELETE-FIXUP are **terminating**: once encountered, they cause the loop to terminate after a constant number of additional operations. For each of the cases of RB-INSERT-FIXUP and RB-DELETE-FIXUP, specify which are terminating and which are not. (*Hint*: Look at Figures 13.5, 13.6, and 13.7.)

We shall first analyze the structural modifications when only insertions are performed. Let T be a red-black tree, and define $\Phi(T)$ to be the number of red nodes in T . Assume that 1 unit of potential can pay for the structural modifications performed by any of the three cases of RB-INSERT-FIXUP.

c. Let T' be the result of applying Case 1 of RB-INSERT-FIXUP to T . Argue that $\Phi(T') = \Phi(T) - 1$.

d. When we insert a node into a red-black tree using RB-INSERT, we can break the operation into three parts. List the structural modifications and potential changes resulting from lines 1–16 of RB-INSERT, from nonterminating cases of RB-INSERT-FIXUP, and from terminating cases of RB-INSERT-FIXUP.

e. Using part (d), argue that the amortized number of structural modifications performed by any call of RB-INSERT is $O(1)$.

We now wish to prove that there are $O(m)$ structural modifications when there are both insertions and deletions. Let us define, for each node x ,

$$w(x) = \begin{cases} 0 & \text{if } x \text{ is red,} \\ 1 & \text{if } x \text{ is black and has no red children,} \\ 0 & \text{if } x \text{ is black and has one red children,} \\ 2 & \text{if } x \text{ is black and has two red children.} \end{cases}$$

Now we redefine the potential of a red-black tree T as

$$\Phi(T) = \sum_{x \in T} w(x),$$

and let T' be the tree that results from applying any nonterminating case of RB-INSERT-FIXUP or RB-DELETE-FIXUP to T .

- f.** Show that $\Phi(T') \leq \Phi(T) - 1$ for all nonterminating cases of RB-INSERT-FIXUP. Argue that the amortized number of structural modifications performed by any call of RB-INSERT-FIXUP is $O(1)$.
- g.** Show that $\Phi(T') \leq \Phi(T) - 1$ for all nonterminating cases of RB-DELETE-FIXUP. Argue that the amortized number of structural modifications performed by any call of RB-DELETE-FIXUP is $O(1)$.
- h.** Complete the proof that in the worst case, any sequence of m RB-INSERT and RB-DELETE operations performs $O(m)$ structural modifications.
- a.** If we insert a node into a complete binary search tree whose lowest level is all red, then there will be $\Omega(\lg n)$ instances of case 1 required to switch the colors all the way up the tree. If we delete a node from an all-black, complete binary tree then this also requires $\Omega(\lg n)$ time because there will be instances of case 2 at each iteration of the **while** loop.
- b.** For RB-INSERT, cases 2 and 3 are terminating. For RB-DELETE, cases 1 and 3 are terminating.
- c.** After applying case 1, z 's parent and uncle have been changed to black and z 's grandparent is changed to red. Thus, there is a net loss of one red node, so $\Phi(T') = \Phi(T) - 1$.
- d.** For case 1, there is a single decrease in the number of red nodes, and thus a decrease in the potential function. However, a single call to RB-INSERTFIXUP could result in $\Omega(\lg n)$ instances of case 1. For cases 2 and 3, the colors stay the same and each performs a rotation.
- e.** Since each instance of case 1 requires a specific node to be red, it can't decrease the number of red nodes by more than $\Phi(T)$. Therefore the potential function is always non-negative. Any insert can increase the number of red nodes by at most 1, and one unit of potential can pay for any structural modifications of any of the 3 cases. Note that in the worst case, the call to RB-INSERT has to perform k case-1 operations, where k is equal to $\Phi(T_i) - \Phi(T_{i-1})$. Thus, the total amortized cost is bounded above by $2(\Phi(T_n) - \Phi(T_0)) \leq n$, so the amortized cost of each insert is $O(1)$.

f. In case 1 of RB-INSERT, we reduce the number of black nodes with two red children by 1 and we at most increase the number of black nodes with no red children by 1, leaving a net loss of at most 1 to the potential function. In our new potential function, $\Phi(T_n) - \Phi(T_0) \leq n$. Since one unit of potential pays for each operation and the terminating cases cause constant structural changes, the total amortized cost is $O(n)$ making the amortized cost of each RB-INSERT-FIXUP $O(1)$.

g. In case 2 of RB-DELETE, we reduce the number of black nodes with two red children by 1, thereby reducing the potential function by 2. Since the change in potential is at least negative 1, it pays for the structural modifications. Since the other cases cause constant structural changes, the total amortized cost is $O(n)$ making the amortized cost of each RB-DELETE-FIXUP $O(1)$.

h. As described above, whether we insert or delete in any of the cases, the potential function always pays for the changes made if they're nonterminating. If they're terminating then they already take constant time, so the amortized cost of any operation in a sequence of m inserts and deletes is $O(1)$, making the total amortized cost $O(m)$.