CPSC 331 — Solutions for Tutorial Exercise #11
Red-Black Trees

In this exercise you were asked to complete a proof of the correctness of the insertion algorithm for red-black trees by confirming that the ***black-heights*** of nodes are still defined after each of the possible adjustments (described near the end of Lecture #11) are made.

Note: In order to simplify wordings, the label that a node has (like α or β) will sometimes be used as the name for the node, itself.

1. You were first asked to consider the effects of the adjustment that is used, during the execution of the `while` loop in the insertion algorithm for a red-black tree, if subcase 1(a) is detected: Both `z` and its parent are ***left*** children and the sibling `y` of the parent of `z` is ***red***, as shown in Figure 1 on page 2. In this case, one adjusts the tree by recolouring the nodes with labels β, γ and δ, and setting `z` to point to its current grandparent. The result is as shown in Figure 2.

   (a) You were asked to explain, briefly, why the black-heights of all nodes in the subtrees T_1, T_2, T_3, T_4 and T_5 are still well-defined, and unchanged by the operation that is carried out.

   Solution: None of these subtrees have been changed by this operation. Thus black-heights of all nodes in these subtrees are well-defined after this operation because these were well-defined before it.

   (b) You were asked to explain why the black-heights of the nodes with labels α, β and δ are still well-defined, and unchanged, as well.

   Solution: The subtrees with each of these nodes as roots have not been changed by this operation either.

   (c) You were asked to explain why the black-height of the node with label γ is increased by one, but still well-defined, as well.

   Solution: Since β and δ were both the children of γ before this operation and both had colour red, they must have had the same black-height before this operation:

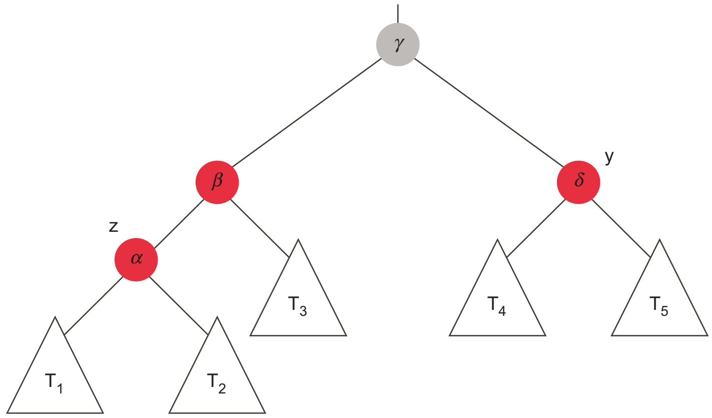

Figure 1: Case 1(a): Tree Before Operation

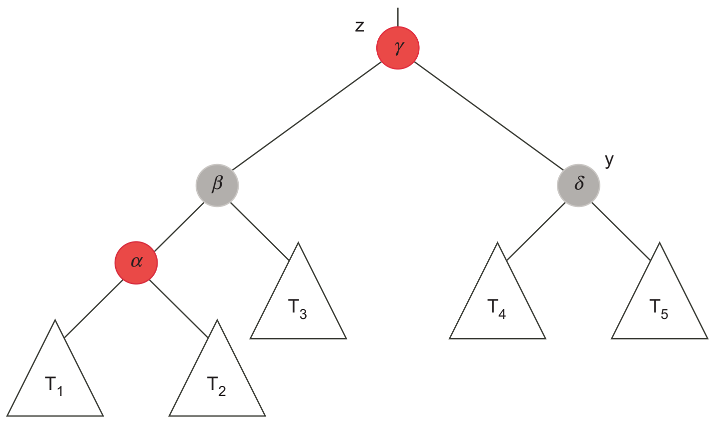

Figure 2: Case 1(a): Tree After Operation

The black-height of γ would not have been well-defined otherwise. Furthermore, since β and γ were ***red*** the black-height of γ must have been equal to the black-height of β before the operation.

After the operation, β and δ are still the children of γ, but they are now ***black*** instead of ***red***. Since every path from γ down to a leaf must continue with either β or δ (but not both) it follows that the black-height of γ has been increased by one.

(d) You were asked to explain why the black-height of ***every other*** node in this red-black tree is also well-defined and unchanged by this adjustment.

Solution: If γ is the root of this tree then there are no other nodes to consider. Otherwise, since γ was originally ***black*** its black-height must have been the same as the black-height of its sibling, if the sibling was ***black***, or it must have been one *less* than the black-height of its sibling if the sibling was ***red***: The black height of the parent of γ would not have been well-defined otherwise.

As noted above, the black-height of γ has been increased by one — but γ now has colour ***red*** instead of ***black*** — so that the number of black nodes (including γ) on any path from γ down to a leaf has not been changed. These are all the same, and they are also equal to the number of black nodes on any path from the *sibling* of γ down to a leaf, including the sibling if the sibling has colour ***black***. It follows that the black-height of the ***parent*** of γ is still well-defined and has not changed.

It is now easy to prove that the black-height of every ***ancestor*** of γ is still well-defined and has not been changed. This could be proved by mathematical induction, specifically, induction on the number of edges on the simple path from γ up to the ancestor in the tree. The standard form of mathematical induction could be used.

The only other nodes that must be considered are nodes that are not in the subtree of γ, and that are also not ancestors of γ. The black-heights of all these nodes are still well-defined and unchanged because the subtrees with these nodes have root were not changed by the operation.

(e) Finally, you were asked to confirm that if z pointed to the only ***red*** node in this tree with a ***red*** parent *before* this adjustment, then z points to the only ***red*** node that might be at the root or have a ***red*** parent, *after* the adjustment, as well.

Solution: Before this operation, the only ***red*** node with a red parent was α. Since α's parent (β) is now ***black***, α is no longer a red node with a red parent .

The only nodes that could *become* red nodes with red parents, as a result of this operation, are either

- red nodes whose colours were not changed, but whose parents' colours have been changed from ***black*** to ***red***, or
- nodes whose colours have been changed from ***black*** to ***red***.

One can see by an examination of the trees in Figures 1 and 2 that there are no red nodes, whose parents' colour has been changed from black to red, at all. There is only one node whose colour was changed from black to red, namely, the node γ (now the value of z).

Thus there is at most one red node with a red parent in the tree, as claimed.

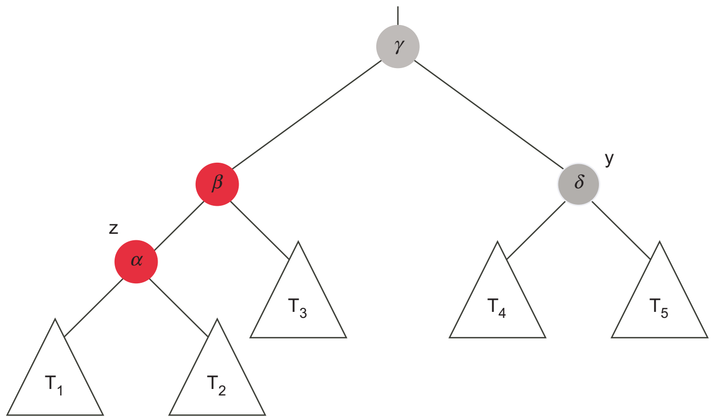

Figure 3: Insertion Case 2: Tree Before Operation

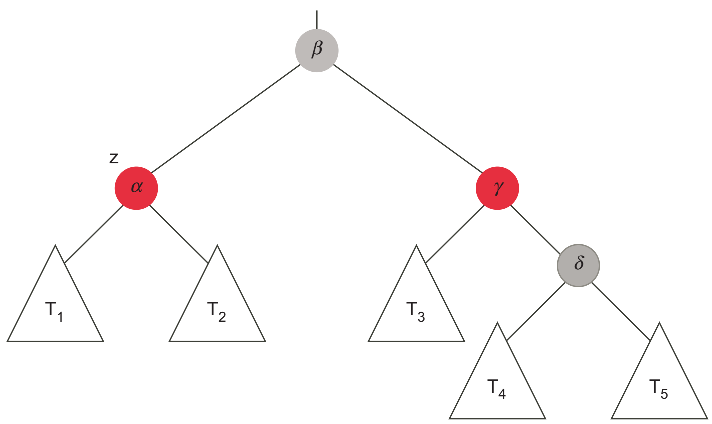

Figure 4: Insertion Case 2: Tree After Operation

2. You were next asked to consider the effects of the adjustment that is used, during the execution of the `while` loop in the insertion algorithm for a red-black tree, if subcase 2 is detected: Both `z` and its parent are ***left*** children and the sibling `y` of the parent of `z` is ***black*** — as shown in Figure 3. In this case, one adjusts the tree by rotating right at the node with label γ and recolouring the nodes with labels β and γ — resulting in the tree shown in Figure 4.

   (a) You were asked to briefly explain, once again, why the black-heights of all nodes in the subtrees T_1, T_2, T_3, T_4 and T_5 are all still well-defined, and unchanged, by this adjustment.

Solution: The answer is the same as the answer for Question 1(a).

(b) Assuming that the black-heights of nodes were all initially well-defined, you were asked to confirm that

$$\mathsf{bh}(\alpha) = \mathsf{bh}(\beta) = \mathsf{bh}(\gamma) = \mathsf{bh}(\delta) + 1$$

before this adjustment.

Solution: The black-heights of α and β are initially the same, as claimed, because α is ***red*** and is a child of β. It must be true that bh$(\beta) =$ bh(γ), initially, because β is ***red*** and is a child of γ. On the other hand, since δ is initially a ***black*** child of γ it must initially be true that bh$(\gamma) =$ bh$(\delta) + 1$. Thus (initially)

$$\mathsf{bh}(\alpha) = \mathsf{bh}(\beta) = \mathsf{bh}(\gamma) = \mathsf{bh}(\delta) + 1$$

as claimed.

(c) You were asked to this information to confirm that the black-heights of α and δ are both well-defined, and unchanged after this operation is carried out.

Solution: The subtrees with roots α and δ have not been changed — so the black-heights of α and δ are well-defined and unchanged as well.

(d) You were asked to use the above information to confirm that the black-height of γ is also well-defined and unchanged by this adjustment.

Solution: Since T_3 is initially the right subtree of the subtree with root β, the number of nodes from the root of T_3 down to any leaf (including the root of this subtree, itself) is equal to the initial black-height of β. This is equal to bh$(\delta) + 1$ both before and after this operation, as noted above. It follows that the black-height of their parent, γ, is well-defined, and equal to bh$(\delta) + 1$ — so that it is unchanged too.

(e) Using *this* information, as well, you were asked to confirm that the black-height of β is still well-defined and also unchanged by this adjustment.

Solution: As noted above, the black-heights of α and γ are still well-defined and have not changed. Since they are now both ***red***, and the children of β, this implies that the black-height of β is still well-defined and that bh$(\beta) =$ bh$(\alpha) =$ bh(γ) – so that the black-height of β is unchanged as well.

(f) You were asked to briefly why the black-height of ***every other*** node in this tree is unchanged by this adjustment.

Solution: The answer here is almost identical to the answer for Problem 1(d).

(g) Finally, note that if z pointed to the only ***red*** node with a ***red*** parent in the tree before the adjustment, then all of the red-black properties are satisfied after it.

Solution: Since the roots of subtrees T_1, T_2, and T_3 all had red parents before this operation and were not the "problem" node z, the roots of all these subtrees must

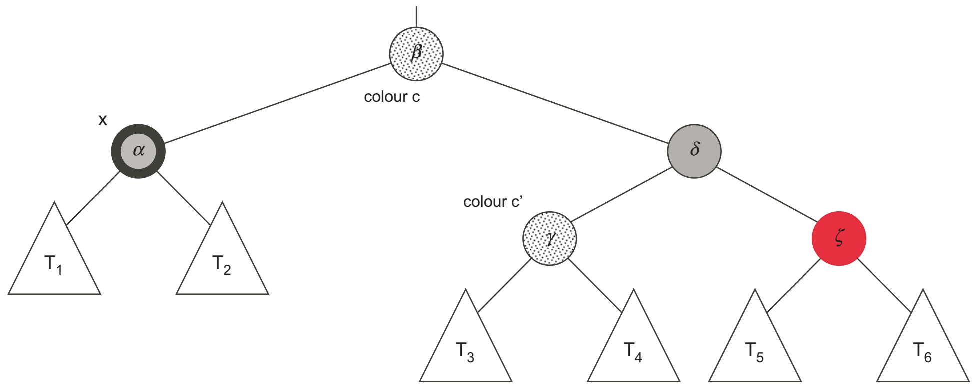

Figure 5: Deletion Case 3(a): Tree Before Operation

be black. Nodes α and γ therefore each have two black children, as required, so there is no node shown in Figure 4 that is a red node with a red parent.

The root of the subtree being shown is still black — so all the other nodes in this tree have the same colour as before, and their *children* have the same colour before. It follows that the red-black properties are satisfied, as claimed.

3. You were next asked to consider the algorithm for the ***deletion*** of a node, storing a given `element` or `key` in a red-black tree, Consider the effects of the adjustment that is used, during the execution of the `while` loop in the deletion algorithm for a red-black tree, if subcase 3(a) is detected, that is, δ is ***black*** and ζ is ***red***, as shown in Figure 5.

   As described in the lecture notes one should proceed by performing a left rotation at β and recolouring α and ζ. The nodes β and δ should switch colours, and `x` should now point to the root of the subtree that has been adjusted — that is, it should point to the node with label δ. The resulting subtree is as shown in Figure 6.

   It can be argued that the black-heights of all of the nodes in subtrees T_1, T_2, T_3, T_4, T_5 and T_6 are all still well-defined, and unchanged, after this adjustment has been made.

   (a) You were asked to confirm that bh$(\beta) =$ bh$(\alpha) + 2 =$ bh$(\delta) + 1 =$ bh$(\zeta) + 1$ ***before*** any adjustments are made.

   Solution: bh$(\beta) =$ bh$(\alpha)+2$ because α is a child of β and the colour of α is initially ***double-black***. bh$(\beta) =$ bh$(\delta) + 1$ because δ is also a child of β and the colour of δ is initially ***black***. Finally, bh$(\delta) =$ bh(ζ), so that bh$(\beta) =$ bh$(\delta) + 1 =$ bh$(\zeta) + 1$, because ζ is a child of δ and the colour of ζ is initially ***red***.

   (b) You were asked to confirm that if the initial colour c' of γ is ***red*** then bh$(\delta) =$ bh(γ) before any adjustments are made.

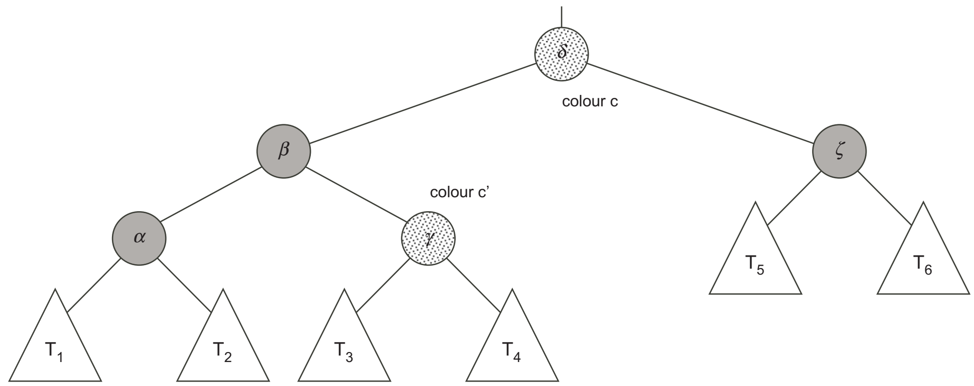

Figure 6: Deletion Case 3(a): Tree After Operation

Solution: γ is a child of δ is ***red***. It follows by the definition of "black-height" that bh(δ) = bh(γ), since every path from γ down to a leaf can be turned into a path from δ down to a leaf by including an edge from δ to γ at the beginning.

(c) You were also asked to confirm that if the initial colour c' of γ is ***black***, then bh$(\delta) =$ bh$(\gamma) + 1$ before any adjustments are made, instead.

Solution: Again, any path from γ down to a leaf can be turned into a path from δ down to a leaf by adding an edge from δ to γ at the beginning. Since γ is ***black*** it follows that bh(δ) = bh$(\gamma) + 1$.

(d) You were asked to use the above to show that bh(α) $=$ bh$(\gamma) - 1$ before any adjustments are made if the initial colour c' of γ is ***red***, and that bh(α) $=$ bh(γ) before any adjustments are made if the initial colour c' of γ is ***black***.

Solution: Since α and δ are initially children of β — whose black-height is initially well-defined — and since α is initially ***double-black***, while δ is initially ***black***, bh(δ) = bh$(\alpha) + 1$; that is, bh(α) = bh$(\delta) - 1$. It follows by the result established in part (b) that bh(α) = bh$(\gamma) - 1$ if γ is initially ***red***, and it follows by the result established in part (c) that bh(α) = bh(γ) if γ is initially ***black***.

(e) You were asked to confirm that the black-heights of α and γ are still well-defined — and unchanged — after this adjustment.

Solution: The black-height of α is still well-defined, and unchanged, because the left and right subtrees of the tree with α as root have not been changed — they are still T_1 and T_2. The black-height of γ is still well-defined and unchanged for virtually the same reason: The left and right subtrees of the subtree with γ as its root are still T_3 and T_4.

(f) You were asked to use this to confirm that the black-height of β is still well-defined, but has been ***decreased by one***, as a result of this adjustment.

Solution: While α is still the left child of β the colour of α has been changed from ***double-black*** to ***black***. Thus the number of black-nodes from β, down to α, and then down to a root has been decreased by one — when β is not included and a double-black node is counted as two black nodes.

As noted above, it was initially true that $\mathsf{bh}(\gamma) = \mathsf{bh}(\delta)$ if the colour c' of γ is ***red***. Since it was initially true that $\mathsf{bh}(\delta) = \mathsf{bh}(\beta) - 1$, as established in part (a), and the black-height of γ is unchanged, as shown in part (e), the black-height of γ is one less than the original black-height of β — and (since γ is ***red***) the number of black nodes on a path from β, to γ and on down to a leaf, is also one less than the original black-height of γ. It follows that the black-height of β is still well-defined after this operation but has been decreased in value by one, as claimed.

On the other hand, if the colour c' of γ was ***black***, then it follows by the results established in parts (d) and (f) that the black-heights of α and γ are the same. Since α and γ are now the children of β, and have the same colour, it follows that the black-height of β is still well-defined — but (as noted above) must now have been decreased in value by one, in this case as well.

(g) You were asked to confirm that the black-height of ζ is still well-defined — and unchanged — as a result of this adjustment.

Solution: The black-height of ζ is still well-defined, and unchanged, because of the left and right subtrees of the subtree with root ζ have not been changed — they are still T_5 and T_6, respectively.

(h) You were asked to use this to confirm that the black-height of δ is well-defined after this adjustment — and that this is equal to black-height that β had ***before*** the adjustment.

Solution: Since β is now the left child of δ and is ***black***, the number of black nodes on any path from δ, down to β, and on to a leaf (not including δ, but including β) is one more than the new black-height of β. By the result established in part (f) this number of black nodes is the same as the original black-height of β.

On the other hand, since ζ is now the right child of δ and its colour is now ***black***, the number of black nodes on a path from δ, down to ζ and on down to a leaf, is one more than the current (and, by (g), initial) black-height of ζ. By the result established in part (a), this is also the same as the original black-height of β.

It follows that the black-height of δ is still well-defined, and is now equal to the original black-height of β, as claimed.

(i) You were asked to use this to confirm that the black-height of the original ***parent*** of β is still well-defined, and unchanged by this adjustment, if β was not originally the root of this red-black tree.

Solution: Notice that the subtree with the former *sibling* of β — which is now the sibling of δ has not been changed, so the number of any black nodes on any path from the parent of δ, down to this sibling, and further on to a leaf (not including the parent) has not been changed. On the other hand, the number of black nodes from the parent down to δ, and further on down to a leaf, is now the same as the number of black nodes from the parent, down to β, and down to a leaf was before the operation (not including the parent, once again, in each case). It follows that the black-height of the parent is well-defined after the operation because it was well-defined before it — and the black-height has not changed.

(j) Finally, you were asked to use this to confirm that the black-heights of all *other* nodes in this red-black tree are still well-defined, and unchanged by this adjustment.

Solution: See the last part of the solution for Problem 1(d): The argument is the same.