

AVL Trees

June 11, 2020

Description

Insertion

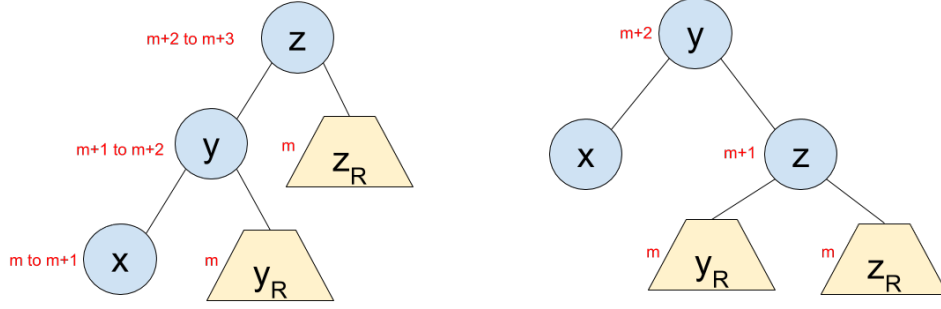
The first part of insertion is the same as inserting into a regular BST. However, inserting may cause one or more nodes in the tree to become imbalanced. Note that when we insert a value, the only nodes whose heights change are the ancestors of the node that we inserted. So, if inserting a node causes imbalance, the inserted value must have at least two ancestors (otherwise the length of the path from the root to the inserted node has increased to at most 1, which cannot cause an imbalance since this would require a height of -1 in the other subtree).

Now, suppose that inserting a node v caused an imbalance. Let z be the first node on the path from v to the root that was imbalanced. Let y be the child of z on that path and x be the grandchild of z on that path. Since z became imbalanced, we know the $h(y)$ increased and therefore the $h(x)$ increased as well. Suppose originally $h(x) = m$. There are 4 cases:

1. LL: y is the left child of z and x is the left child of y .

Since $h(x)$ was m , that means $h(y_R)$ be $m-1, m$, or $m+1$, since y was balanced. After the insertion $h(x) = m+1$. Now, y is still balanced, so $h(y_R) \neq m-1$. Since $h(y)$ increased, $h(y_R) \neq m+1$. Therefore, $h(y_R) = m$. Thus, $h(y)$ increased from $m+1$ to $m+2$.

Now, since the $h(y)$ was $m+1$, $h(z_R)$ could be $m, m+1$, or $m+2$. Since z is now imbalanced, we know $h(z_R) = m$. Therefore, $h(z)$ increased from $m+2$ to $m+3$. After the rotation, we can see that all the nodes are balanced. Also note that the height of the subtree is $m+2$ both before the insertion and after the



rotation. This means that since all ancestors of this node were balanced before the insertion, they will all be balanced again after the rotation. Therefore, when inserting, we do not need to check any ancestor nodes after the initial rotation.

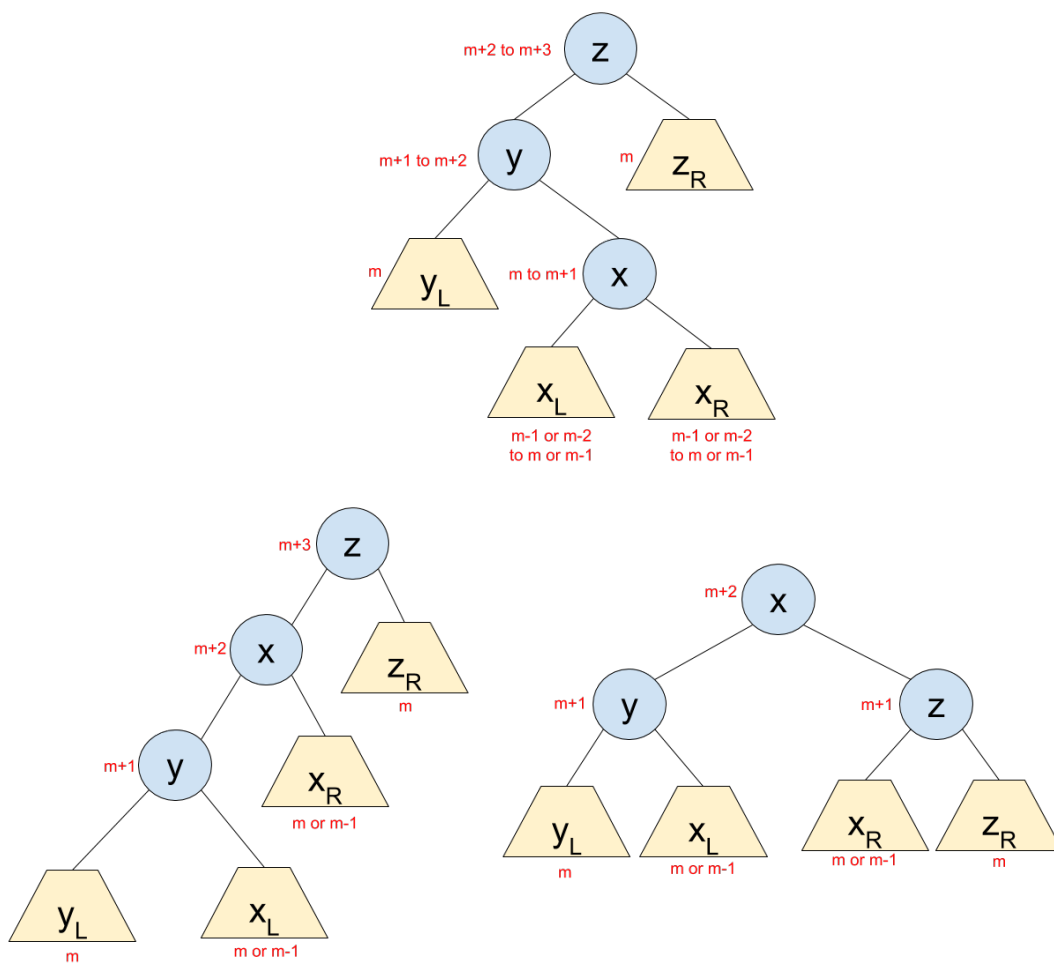
2. RR: y is the right child of z and x is the right child of y . This is analogous to the previous case.
3. LR: y is the left child of z and x is the right child of y .

Since $h(x)$ was m , that means $h(y_L)$ can be $m - 1$, m , or $m + 1$, since y was balanced. After the insertion $h(x) = m + 1$. Now, y is still balanced, so $h(y_L) \neq m - 1$. Since $h(y)$ increased, $h(y_L) \neq m + 1$. Therefore, $h(y_L) = m$. Thus, $h(y)$ increased from $m + 1$ to $m + 2$.

Now, since the $h(y)$ was $m + 1$, $h(z_R)$ could be m , $m + 1$, or $m + 2$. Since z is now imbalanced, we know $h(z_R) = m$. Therefore, $h(z)$ increased from $m + 2$ to $m + 3$.

Since $h(x)$ was m , that means one of $h(x_L)$ and $h(x_R)$ was $m - 1$ before the insertion and the other was $m - 1$ or $m - 2$. After the insertion, x is still balanced, so one of them is m and the other is m or $m - 1$. (We can actually show that they must both be $m - 1$ before, but it is not important for this proof.)

After the double rotation, we can see that all the nodes are balanced. Also note that the height of the subtree is $m + 2$ both before the insertion and after the rotation, which implies the same as before.



4. RL: y is the right child of z and x is the left child of y . This is analogous to the previous case.

Time Complexity