Height Balanced Binary Trees

Case where for each ~~sch~~ node, the height of each of its subtrees differ by at most 1

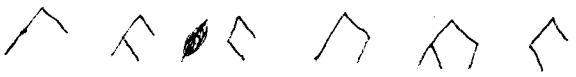
Examples /

} all nodes with ~~an empty~~ subtree are ~~always~~ in the last 3 levels of the tree

How many different height balanced trees are possible for a given height

Height 1 - 1 •

2 - 3 / \ ^

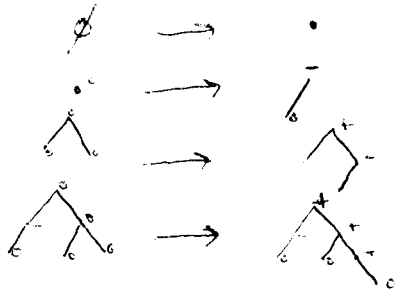
3 - 15 

in general $N(1) = 1$, $N(2) = 3$ $N(n) = N^2(n-1) + 2N(n-1)N(n-2)$

4 - 315 5 - 108675

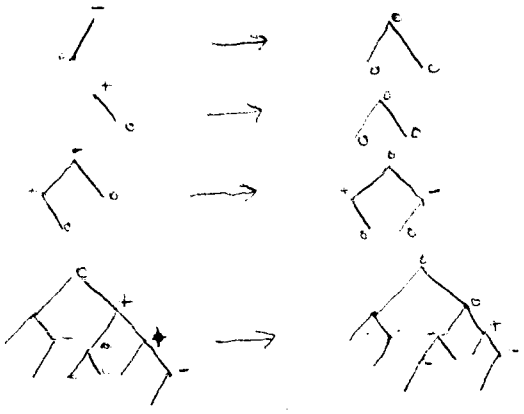
Insertion

Case 1 - insertion with no pivot node



This insertion involves walking backwards up the tree, changing the balance for each node from 0 to non zero depending on the direction ~~from~~ which we are coming

Case 2 - insertion ~~with~~ to the short side of a pivot node



This insertion involves walking backwards up the tree, changing the balance for each node from 0 to non zero depending on the direction from which we are coming, and Stops when we hit the pivot node, changing its balance to zero.

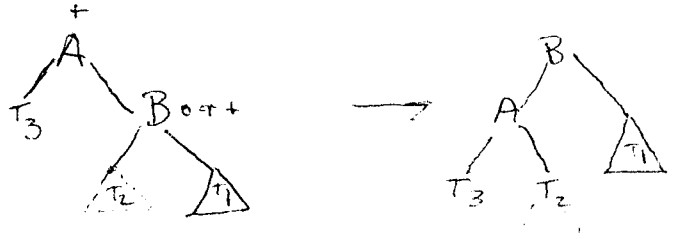
. Summary of Single Rotation Clockwise:

Use it if coming up on a − node (A) from the right and the left child (B) of that pivot node has polarity − or 0. After the rotation, polarity is determined by polarity of B before the rotation:

B before	A after	B after
−	0	0
0	−	+

Note: if B before is −, then we have shortened the tree and we need to proceed back up!

Counterclockwise is the mirror image, used if coming up on a + node (A) from the left and the right child (B) of that pivot node has polarity + or 0.



B before	A after	B after
+	0	0
0	+	−

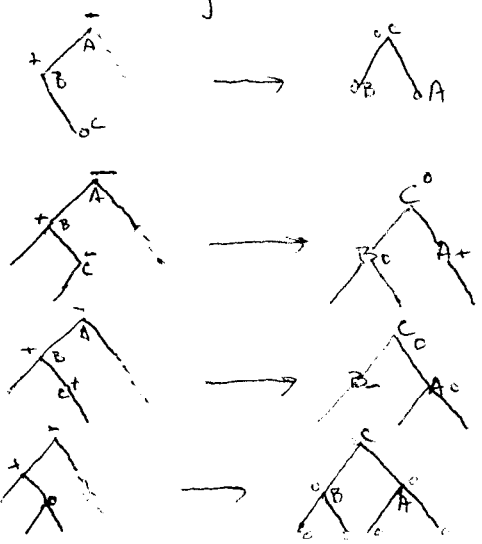
if B before is +, then we have shortened the tree and need to proceed back up

The remaining cases are double rotations, e.g.

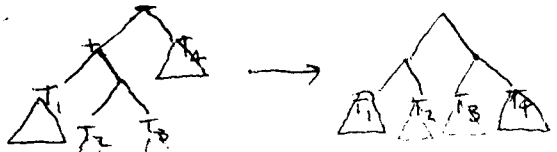
Clockwise, used if coming up on a − node (A) from the right and the left child (B) of that pivot node has polarity +

After the rotation, C is always balanced.

The polarity of A and B depends on the polarity of C before the rotation



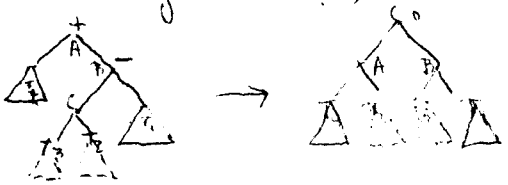
C before	A after	B after
+	0	−
0	0	0
−	+	0



T_1 and T_4 are the same height

T_2 and T_3 may both be that height or one of them (but not both) may be one smaller (except in case T_1 and T_4 are null)

Counter Clockwise is the mirror image, used if coming up on a + node (A) from the left and the right child (B) of that pivot node has polarity −.

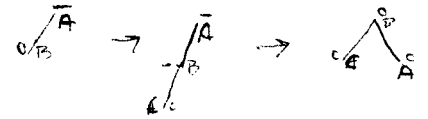


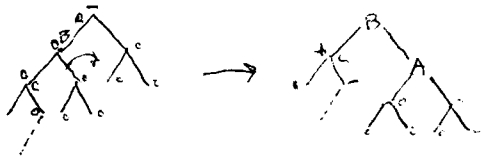
The polarity of A and B depends on the polarity of C before the rotation

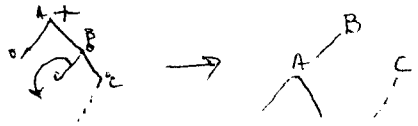
C before	A after	B after
+	−	0
0	0	0
−	0	+

Case 3 - insertion to the long side of a pivot node

Subcase 1 - Single rotation





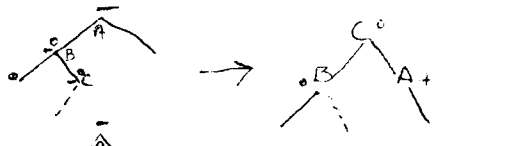


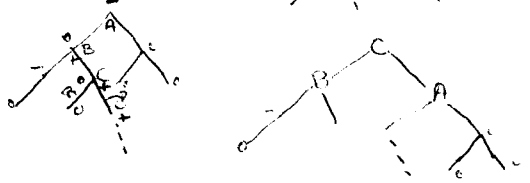
Single rotation is identified when you've come up in the same direction on the two steps before the pivot

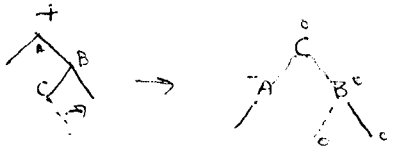
Walk up the tree in the usual manner changing bfs to non-zero until you hit the pivot. Then do the rotation. After the rotation, A and B are always balanced.

There is also the counter-clockwise version

Subcase 2 - Double rotation





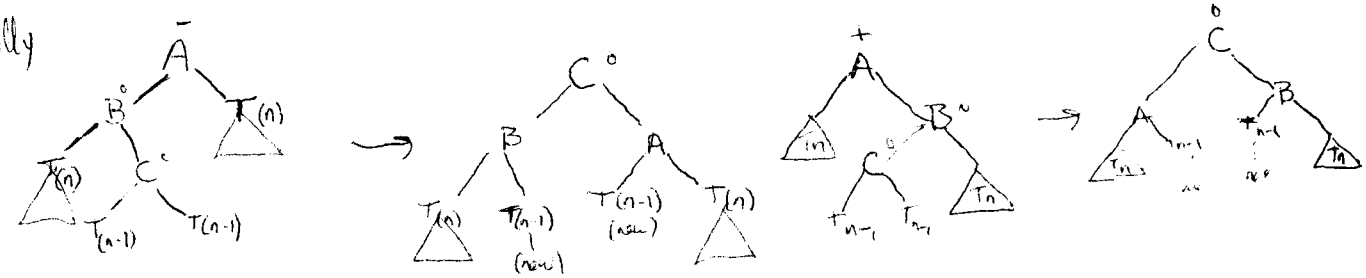


Double rotation is identified when you've come up in ~~the same~~ different directions on the two steps before the pivot

After the rotation, C is always balanced however, the case for A and B varies

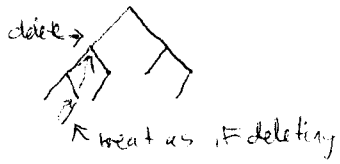
If C is the new node, A and B will be balanced However if C is not the new node, then it depends which side of C the new node was attached to. For clockwise rotation (pivot -) C+ ⇒ A balanced, B -; and C- ⇒ A+, B balanced. For CCW rotation, pivot + C+ ⇒ B balanced and A-, C- ⇒ A balanced, B+

Generally

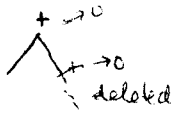


Deletion

We can consider the case of deleting a leaf, because any deletion can be changed to deleting a leaf by exchanging the largest node on the left subtree:



In general, we retrace the deleted node back up the tree, adjusting the balance as we go. If we're coming up from the right, we decrement the balance:



And if were coming up from the left we increment

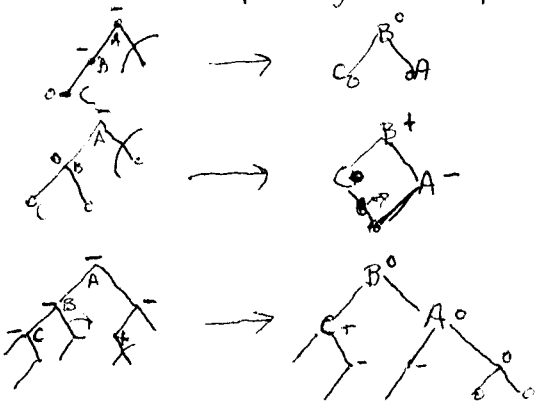
If we come to a node which is balanced, we adjust the balance and stop:
otherwise keep going

Otherwise if we change a balance to zero, keep going, as

The trouble starts if we're coming up from the right on a node which is −, or from the left on a node which is +. Then we have to rotate the tree on that pivot node.

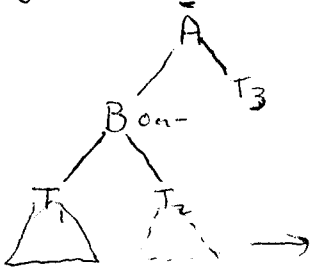
Single rotation clockwise

We can use single rotation if the tree opposite the pivot node is balanced or has the same polarity as the pivot node:

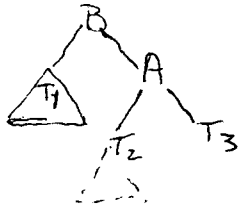


We have to be careful of the polarities when done. This is determined by the polarity of B

In general we have



T_3 is exactly one shorter than T_1. T_2 may be equal might to either T_2 or T_1.



Insertion flow chart

Insert down to leaf level, then walk back up the tree:

If coming up from the right

If the node is balanced, set the balance to + and continue
If the node is − set the balance to 0 and stop
If the node is +, ~~examine the polarity~~
If you've come up from the right on the previous move, single rotate, and set the balance of A and B to zero
If you've come up from the left on the previous move, double rotate, and set A and B according to C's polarity

C before	A after	B after
+	−	0
0	0	0
−	0	+

then set C to 0.

If coming up from the left
If the node is balanced set the balance to − and continue
If the node is + set the balance to 0 and stop
If the node is −
If you've come up from the left on the previous move, single rotate and set the balance of A and B to 0
If you've come up from the right, ~~single rotate~~ double rotate and set A & B according to C's polarity

C before	A after	B after
−	+	0
0	0	0
+	0	−

Deletion flow chart

If not a ~~root~~ leaf node, exchange it with left subtree's right most node, which is either a leaf node or has only one left subnode. Exchange with this node and then exchange with the left subnode. Free the deleted node.

Proceed up the tree.

IF coming up from the right (Dir == 1)

if the node is balanced, set the balance to ~~zero~~ and return

IF the node is + set the balance to 0 and continue

IF the node is −, examine the polarity of the left subtree. (P_3) (B)

IF it is −, single rotate clockwise and set A & B to zero balance

IF it is 0, single rotate clockwise and set A to −, B to +

IF it is + examine the polarity of its right subtree (P_4)

if it is +, double rotate and set A to C, (P_2 was −) B to − (P_3 was +)

IF it is 0 double rotate and set A to C, B to 0

A to + 0

−

IF coming up from the left (Dir == −1)

IF the node is balanced, set the balance to + and return

IF the node is − set the balance to 0 and continue up

IF the node is + (P_2) examine the polarity of the right subtree (P_3)

IF it is + (P_2), single rotate CCW and set A (P_2) & B (P_3) to zero balance

IF it is 0 single rotate CCW and set A to +, B (P_3) to −

IF it is − examine the polarity of the ~~right~~ left subtree

IF it is − double rotate and set A (P_2) to 0 B (P_3) to +

0 0

− 0

0

+

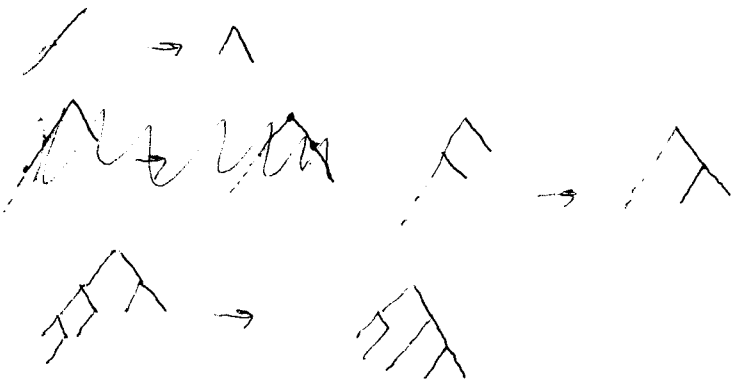
Test cases

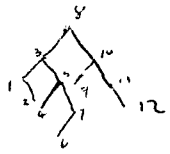
Single node insertion and deletion

Insertion with no pivot node

Insertion to right side of pivot node

Single rotation both dir





Distribution of trees

Height	Total trees	Number of nodes
1	1	1
2	3	2: 2 3: 1
3	15	4: 4 5: 6 6: 4 7: 1
4	315	7: 16 8: 32 9: 44 10: 60 11: 70 12: 56 13: 28 14: 8 15: 1
5	108675	12 128 13 448 14 864 15 1552 16 2720 17 4288 18 6312 19 9004 20 11992 21 14372 22 15400 23 14630 24 11968 25 8104 26 4376 27 1820 28 560 29 120 30 16 31 1

10000 ??