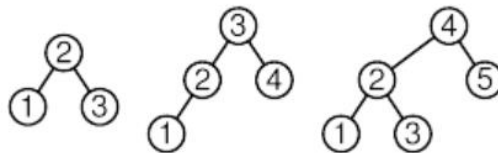


Recitation 6 – AVL

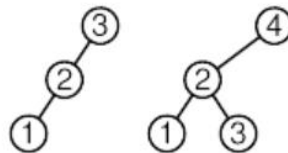
AVL Properties

- (1) Is a BST
- (2) The difference of height between any left and right subtree can be at most 1

AVL examples

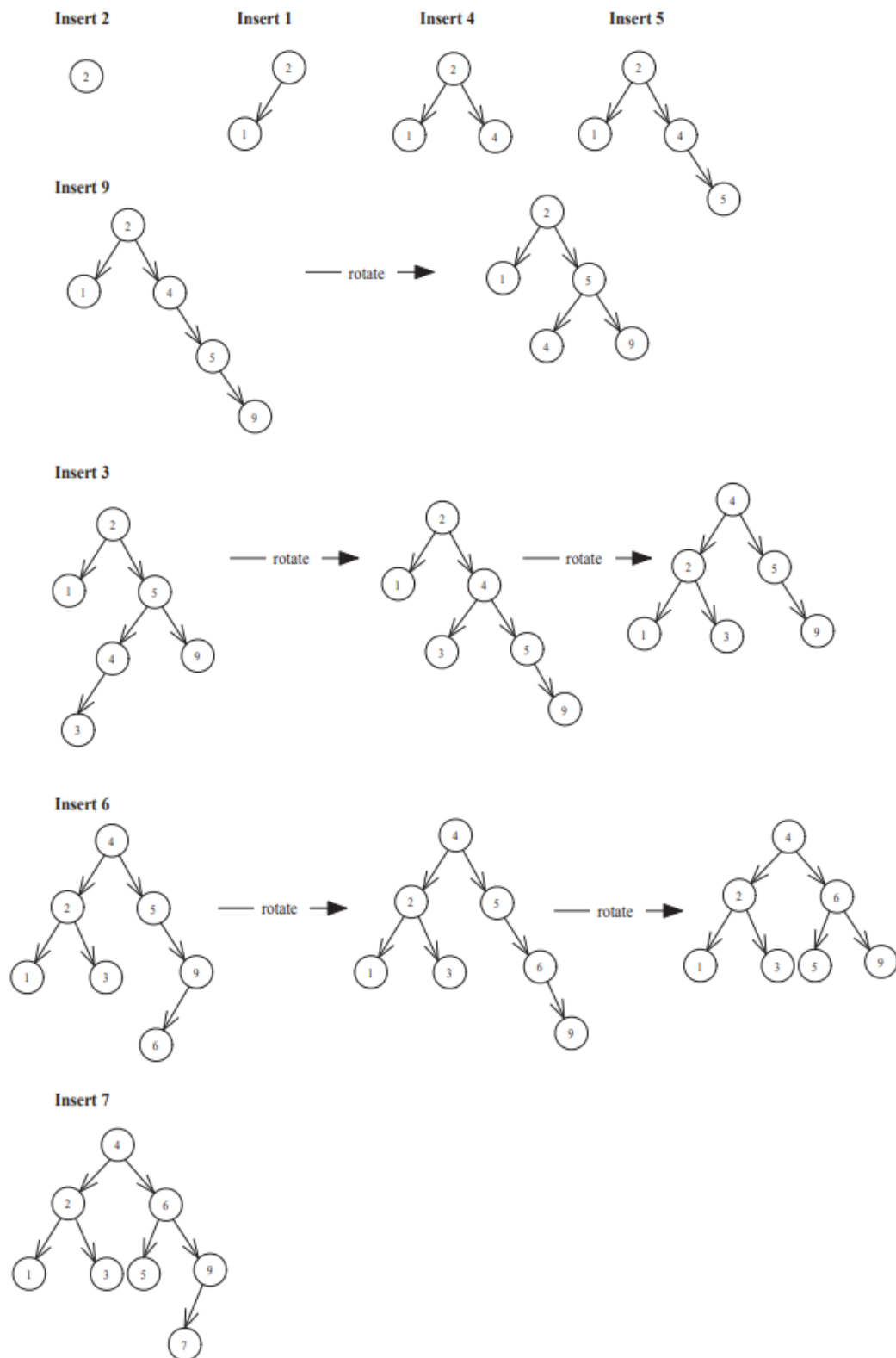


BST but not AVL



Left Left Case	Right Right Case	Left Right Case	Right Left Case
Right Rotation	Left Rotation	Left Rotation	Right Rotation
		Right Rotation	Left Rotation

4.19 Show the result of inserting 2, 1, 4, 5, 9, 3, 6, 7 into an initially empty AVL tree



4.18

(a) Give a precise expression for the minimum number of nodes in an AVL tree of height h .

Let $S(h)$ be the minimum number of nodes in an AVL tree T of height h . The subtrees of an AVL tree with minimum number of nodes must also have minimum number of nodes. Also, at least one of the left and right subtrees of T is an AVL tree of height $h - 1$. Since the height of left and right subtrees can differ by at most 1, the other subtree must have height $h - 2$. Then, we have the following recurrence relation:

$$S(h) = S(h - 1) + S(h - 2) + 1$$

We also know the base cases: $S(0) = 1$ and $S(1) = 2$

Then the sequence looks like: 1, 2, 4, 7, 12, 20, 33, ...

Recall Fibonacci sequence $F(h)$: 0, 1, 1, 2, 3, 5, 8, 13, 21, ...

It seems we have $S(h) = F(h + 3) - 1$. Let us prove this by induction.

Base cases:

$$S(0) = 1, F(3) = 2 \text{ so } S(0) = F(3) - 1$$

$$S(1) = 2, F(4) = 3 \text{ so } S(1) = F(4) - 1$$

Induction hypothesis:

Assume that $S(h) = F(h + 3) - 1$ holds for $h = 1, 2, \dots, k$

Inductive step:

Prove that it is also true for $h = k + 1$

$$\begin{aligned} S(k + 1) &= S(k) + S(k - 1) + 1 \\ &= F(k + 3) - 1 + F(k + 2) - 1 + 1 \\ &= F(k + 4) - 1 \end{aligned}$$

Having proved the identity for $k + 1$, inductively applying same procedure we can prove upto ∞

(b) What is the minimum number of nodes in an AVL tree of height 15?

$$S(15) = F(18) - 1 = 2583$$

4.25

(a) How many bits are required per node to store the height of a node in N-node AVL tree?

In an N-node AVL tree the depth is $O(\log(N))$. To represent an integer range of M we need $O(\log(M))$ bits. Notice that $M \approx \log(N)$ here. Therefore the result is $O(\log(\log(N)))$

As an example, suppose we have an AVL tree with 256 nodes. The depth is around $\log(256) = 8$. To represent upto 8 we need around $\log(8) = 3$ bits.

(b) What is the smallest AVL tree that overflows an 8-bit height counter?

Assuming unsigned data type we can represent $2^8 = 256$ depth values. Then an AVL tree with more than 2^{256} nodes, or equivalently that has a height bigger than 256, will overflow the counter.

References:

CSE 100 – Homework 1- UCSD CSE, <https://cseweb.ucsd.edu/classes/su05/cse100/cse100hw1.pdf>, 11.25.2018

Insertion in AVL, <https://www.includehelp.com/ds/insertion-in-avl-tree-using-c-program.aspx>, 11.25.2018