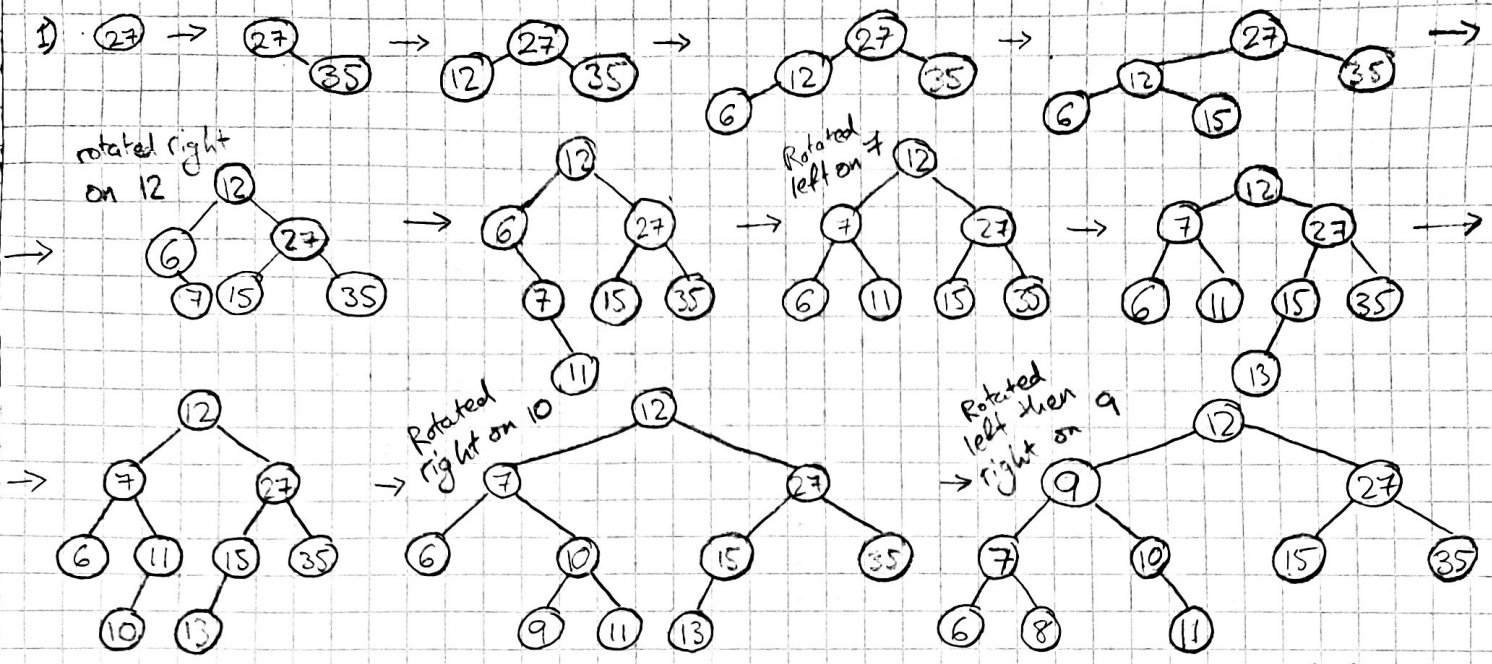
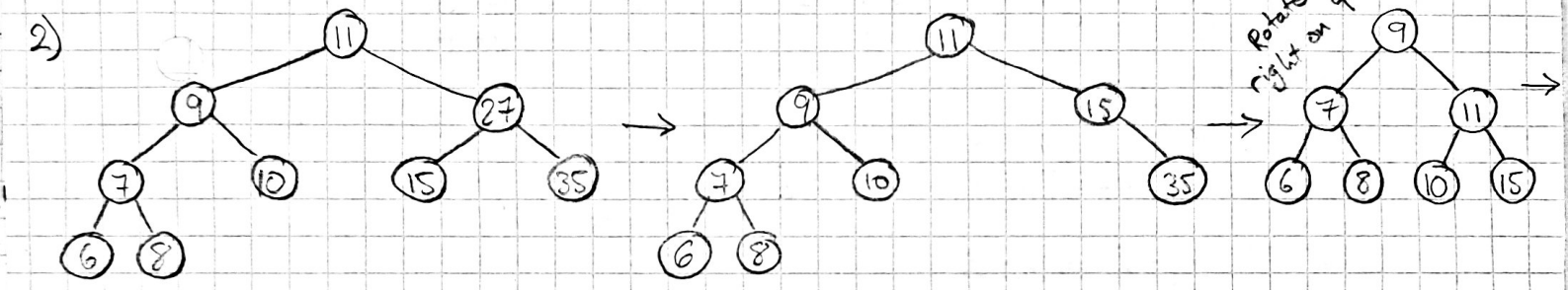
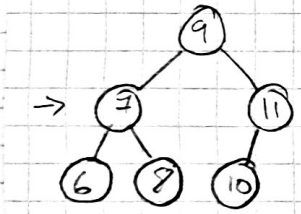
1)

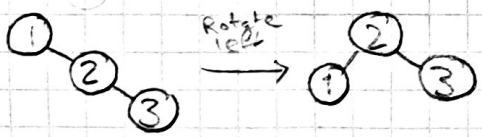


2)



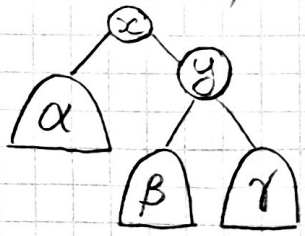


3) Disproof by counterexample: let $n=3$ and the set of numbers be $\{1,2,3\}$. If inserted without rotating, the tree would not be balanced an would require a left rotation on the 2. (see diagram below)



$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} * 1\ 2\ 3$$

4) After correction, the tree will look like this:



$$S_y = S_\beta + S_\gamma + 1$$
$$S_x = S_\alpha + S_y + 1$$

5)
```
node* rotateRL(node* p){
    node* n1 = p;
    node* n2 = n1->right;
    node* n3 = n2->left;

    n2->left = n3->right;
    if(n2->left) { n2->left->parent = n2; }

    n1->right = n3->left;
    if(n1->right) { n1->right->parent = n1; }

    n3->left = n1
    n1->parent = n3

    n3->right = n2
    n2->parent = n3

    return n3;
}
```

6) Maximum nodes in AVL tree of height h is acheived in a full binary tree.
In a full binary tree, each level l has 2^l nodes, so in h levels, there are:
$\sum_{l=0}^{l=h} 2^l$ nodes.

2^l is a geometric series with $a=1$ and $r=2$.
Sum of first h terms of the geometric series $= \frac{1(1-2^h)}{1-2} = 2^{h+1}-1$

7) Yes. Start from the lowest leaves traversing the tree upwards, rotating any unbalanced nodes with a balance factor ± 2. If a balance factor >2 or <-2 is found traverse the tree down from the unbalanced node down the other direction until a ± 2 is found, balance that, then traverse upwards from there. Repeat until balanced