4999406

Certification: I hereby certify that this work is my own and none of it is the work of any other person.

Problem 3:

a) AVL-Tree Insertion-Order Set: {10, 20, 15, 25, 30, 16, 18, 19}

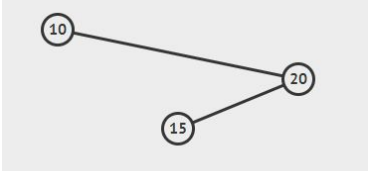
Insertion of 10:

The tree was initially empty so 10 becomes the root.

Insertion of 20:

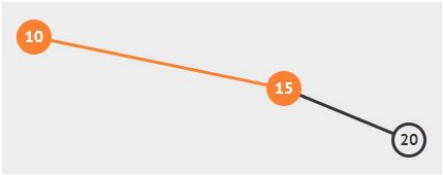
20 is greater than 10 so it becomes its greater child. No need to balance as the height difference between the current two right and left subtrees is 1.

Insertion of 15:



15 is greater than 10 and less than 20, so it becomes the lesser child of 20. This causes the tree to become unbalanced as the difference between the left and right subtrees is now greater than 1.

Balance of {10, 20, 15}:
Right rotation of 15 to 20's position.

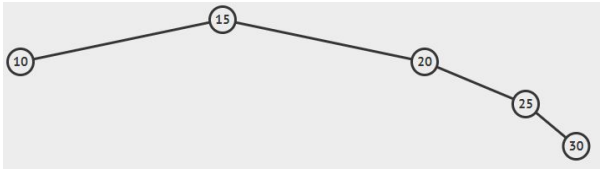


Left rotation of 15 to 10's position. (Result)

Insertion of 25:

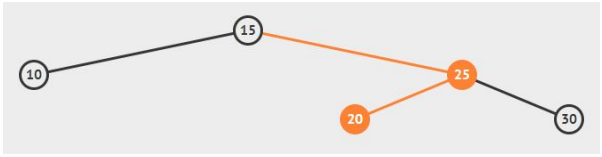
Tree is balanced.

Insertion of 30:



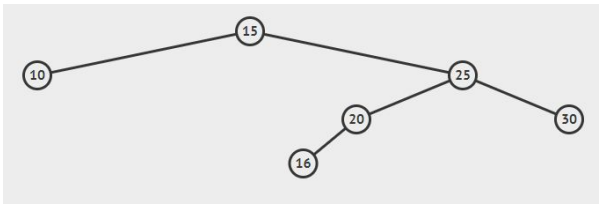
Tree is not balanced, difference between subtree depth of 10 and 20 is greater than 1.

Balance of {10, 20, 15, 25, 30}:
Left rotation of 25 to 20's position.



Tree is now balanced.

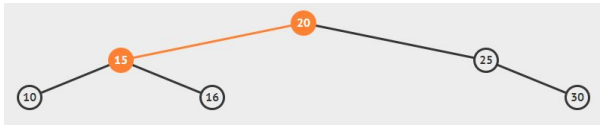
Insertion of 16:



Tree is not balanced, 15's subtree height difference is greater than 1.

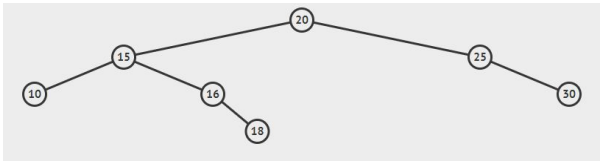
Balance of {10, 20, 15, 25, 30, 16}:
Right rotation of 20 to 25's position.

Left rotation of 20 to 15's position.



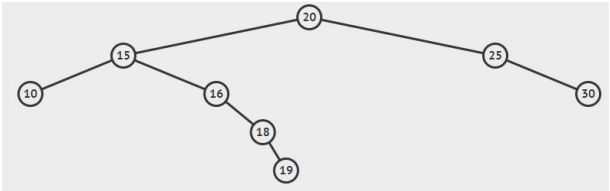
Tree is now balanced.

Insertion of 18:



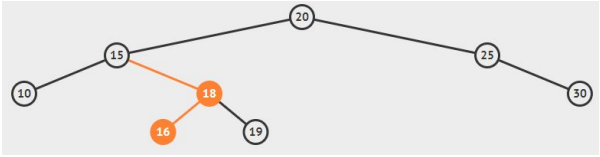
Tree is balanced.

Insertion of 19



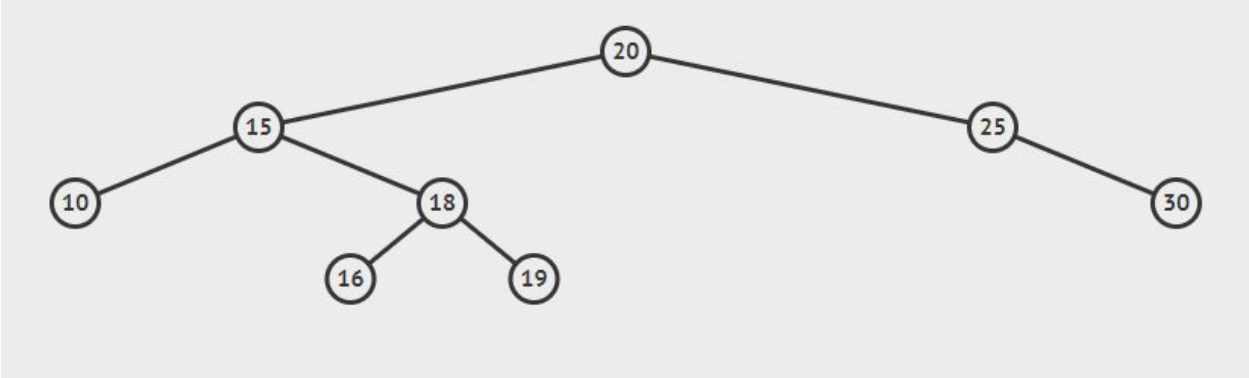
Tree is not balanced. 16's subtree height difference is greater than 1.

Balance of {10, 20, 15, 25, 30, 16, 18, 19}
Left rotation of 18 to 16's location.



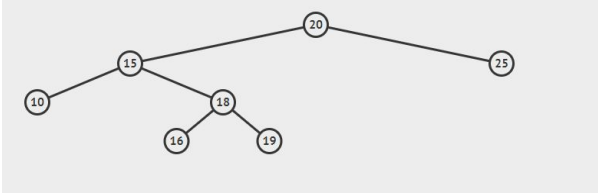
Tree is now balanced.

Final Result:



b) Deletion of element 30 from resulting AVL-Tree from the previous exercise.

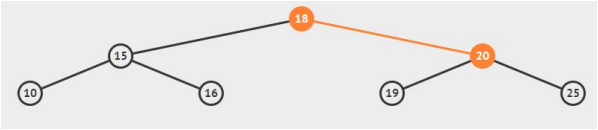
Deletion of 30:



Tree is not balanced. Height difference of 20's subtree's is larger than 1.

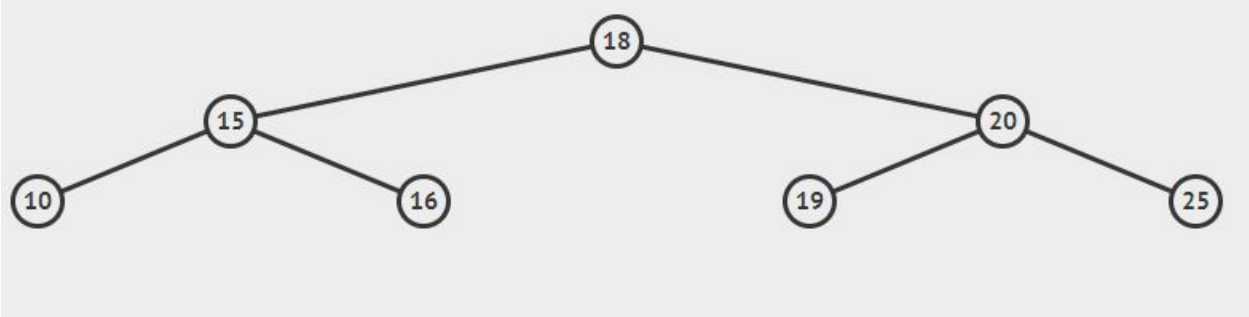
Balance of {10, 20, 15, 25, 16, 18, 19}:
Left rotation of 18 to 15's position.

Right rotation of 18 to 20's position.



Tree is now balanced.

Result:



c) AVL-Tree Insertion-order Set: {1, 2, 3, 4, 5, 6, 7}

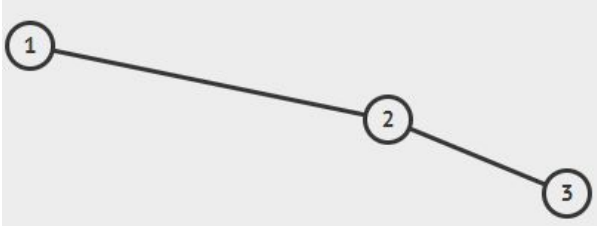
Insertion of 1:

1 is the first element of the tree, so it is inserted as the root.

Insertion of 2:

Tree is balanced.

Insertion of 3:



Tree is unbalanced. Subtrees of 1 height difference is greater than 1.

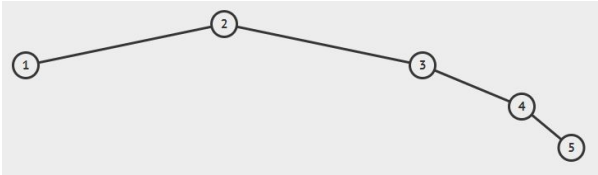
Balance of {1, 2, 3}:
Left rotation of 2 to 1's position.

Tree is balanced.

Insertion of 4:

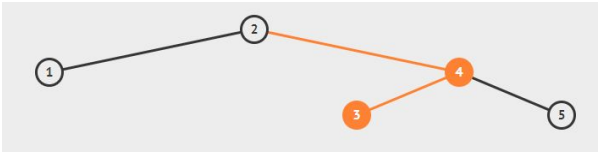
Tree is balanced.

Insertion of 5:



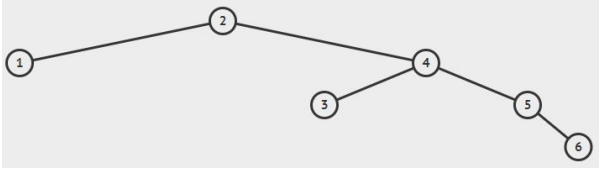
Tree is unbalanced. 3's subtree height difference is greater than 1.

Balance of {1, 2, 3, 4, 5}:
Left rotation of 4 to 3's position.



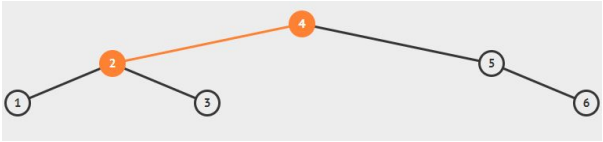
Tree is balanced.

Insertion of 6:



Tree is unbalanced. 2's subtrees height difference is greater than 1.

Balance of {1, 2, 3, 4, 5, 6}:
Left rotation of 4 to 3's position.

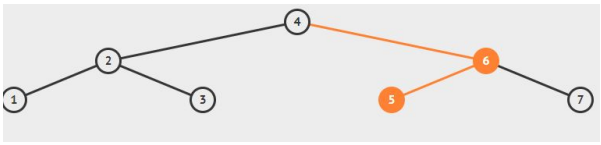


Tree is balanced.

Insertion of 7:

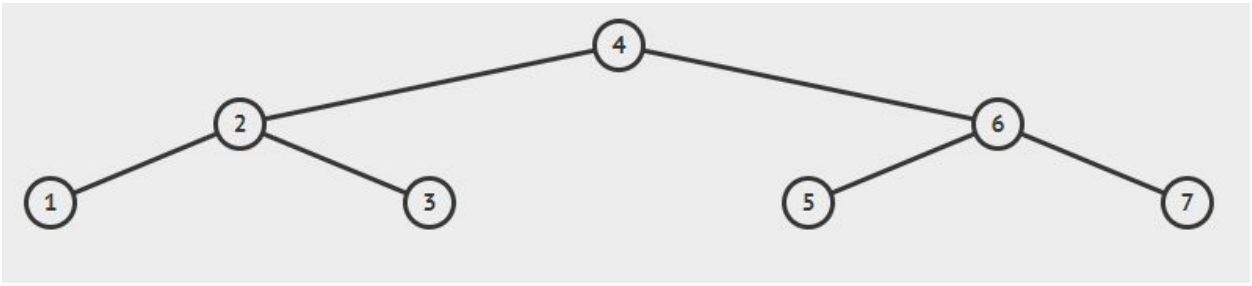
Tree is unbalanced. 5's subtrees height difference is greater than 1.

Balance of {1, 2, 3, 4, 5, 6, 7}:
Left rotation of 6 to 7's position.



Tree is not balanced.

Result:

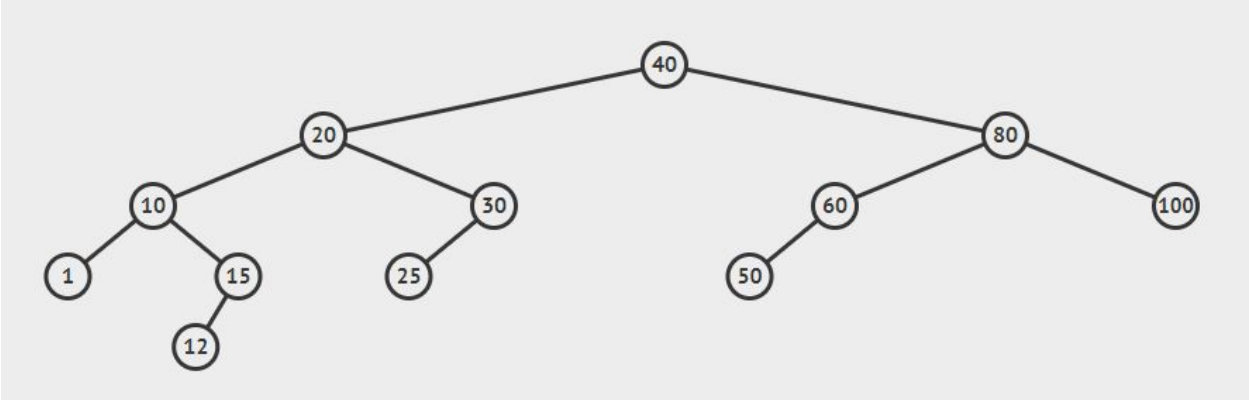


d) Minimum possible nodes for a AVL-Tree of height 4 with diagram.

NodeTotal(height) = NodeTotal(height-1) + NodeTotal(height-2) + 1
= NodeTotal(3) + NodeTotal(2)+1
= (NodeTotal(2)+NodeTotal(1)+1) + (NodeTotal(1)+NodeTotal(0)+1) +1
= (((NodeTotal(1) + NodeTotal(0) + 1) + NodeTotal(1) + 1) + (NodeTotal(1)+NodeTotal(0)+1)
+1
=((2+1+1)+2+1)+2+1+1+1
= 4+2+1+2+1+1+1
= 6+2 +4
= 8+4
= 12

Minimum nodes required: 12

Diagram that fits criteria:

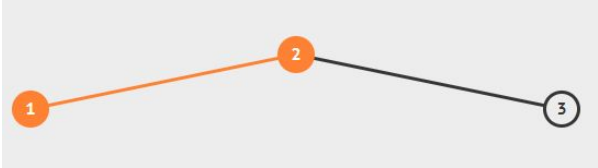


e) AVL-Tree insertion-order set: {1, 2, 3, 8, 6}. State number of rotations and what were the types. (Double rotations count as one rotation.

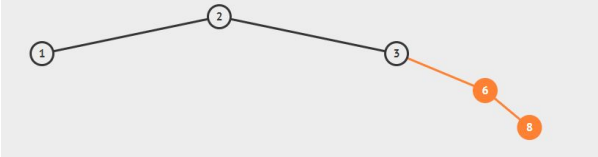
There are a total of 2 rotations. One left rotation, and one right-left rotation.

Rotations:

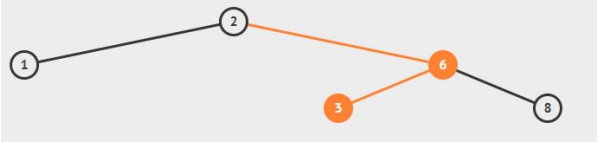
Left Rotation of 2 to 1's position.



Right rotation of 6 to 8's position.



Then a left rotation of 6 to 3's position to complete a right-left rotation.



Result: