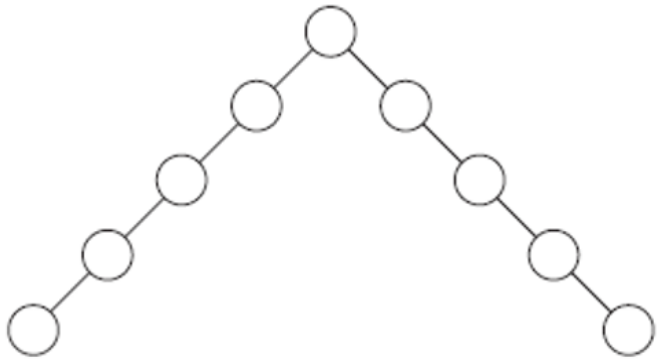
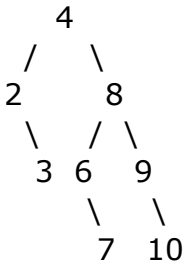
AVL (Adelson-Velsky and Landis) Tree

1. What is an AVL Tree?
   It is a _____ BST, where the height of the left and right subtree can differ by at most _____ for each node **(recursively)**.

1. (T/F) Below is an AVL tree



2. (T/F) Below is an AVL tree



3. (T/F) Inserting a new node to AVL tree following the BST constraint can violate the balance condition. (Consider inserting 11 to above tree)

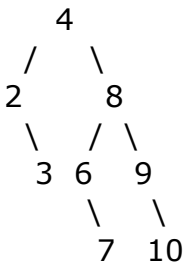
4. (T/F) Removing a node in AVL tree following the BST constraint can violate the balance condition. (Consider removing 3 from above tree.)

5. When the balance condition is violated, we can fix it by performing either _____ or _____
   at the node with the _____ depth experiencing the imbalance.

(Lowest depth means the node farthest from the _____.)

Why?

ex)



Is this AVL? Let's insert 11.

Both node ___ and node ___ are experiencing the imbalance.
Which one should we fix (rotate)? Why?

Rotate 4 vs Rotate 9.

8. Let's start with a balanced AVL tree. And say we insert a node. After the insertion, we have imbalanced AVL. Let's call the node with the lowest/deepest depth experiencing imbalance **t**. We know **t**'s right subtree and left subtree differ height by _____. (Why?)

There are only four cases such violation to balance condition can occur:

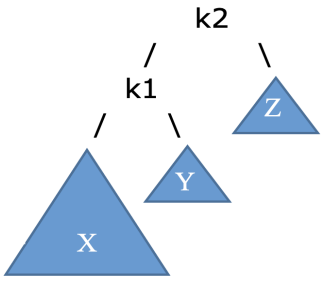
Case 1) An insertion to the left subtree of the left child of t.
Case 2) An insertion to the right subtree of the left child of t.
Case 3) An insertion to the left subtree of the right child of t.
Case 4) An insertion to the right subtree of the right child of t.

9. When the insertion occurs on the outside (such as case 1 and 4), the balance condition can be restored by performing a **_____ rotation**. When the insertion occurs on the *inside* (such as case 2 and 3), the balance condition can be restored by performing **_____ rotations**.

10. **A single rotation with the left child (balancing act for case 1)**

Initial state (needs rebalancing at node k2)

Note that k1's data < k2's data



—>(draw after rotation)

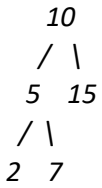
X < k1 < Y < k2 < Z

Single rotation at node k2 with its left child k1

Imaging "lifting" up k1 so that k2 becomes the right child of k1.

Then where should Y go? Y becomes a _____ of k2. This preserves the BST property.

An example of a single rotation with the left child



(Draw the result after inserting 1 with - imbalance!)

We need to do a single rotation at node _____ with its left child.

Which node to "lift" up?

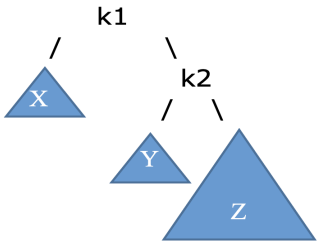
Where should node 7 go?

(Draw the result after performing the single rotation)

11. **A single rotation with the right child (balancing act for case 4)**

Initial state (needs rebalancing at node k1)

Note k1's data < k2's data



—>(draw after rotation)

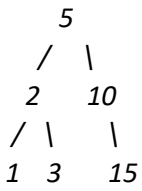
X < k1 < Y < K2 < Z

Single rotation at node k1 with its right child k2

(Imaging "lifting" up k2 so that k1 becomes the left child of k2. Where to put Y? Y becomes a _____ of k1. This is to preserve the BST property.)

An example of a single rotation with the right child

(Given below AVL tree)



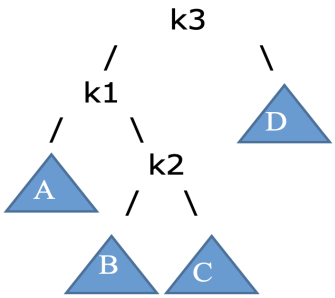
(Draw the result after inserting 20 - imbalance)

Now, we need to do a single rotation at node _____ with its right child.

Which node to "lift" up?

(Draw the result after performing the single rotation)

12. **Double rotation with the left child (balancing act for case 2)**
Initial state (needs rebalancing at node k3)
Note that k1's data < k2's data < k3's data.



A < k1 < B < k2 < C < k3 < D
Trick: Identify kink "k3-k1-k2" and lift "k2" twice.

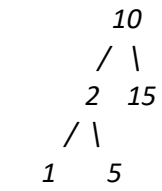
After the first single rotation at k1 (k3's left child) with its right child k2

As you can see, after the first rotation the "inner" problem becomes the "outer" problem, which can be fixed by another rotation.

After the second single rotation at k3 with its left child k2

An example of double rotation with the left child

(Given below AVL tree)



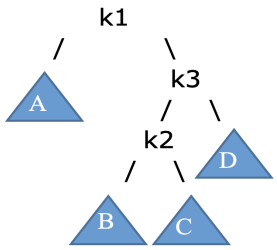
(Draw the result after inserting 3 - imbalance!)

Now, we need to perform double rotation at node _____ .

(Draw the result after the first rotation)

(Draw the result after the second rotation)

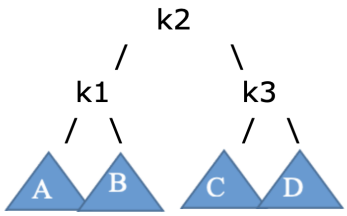
13. **Double rotation with the right child (balancing act for case 3)**
Initial state (needs rebalancing at node k1, let's call k1's right child k3 and k3's left child k2) Note that k1's data < k2's data < k3's data.



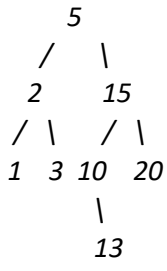
A < K1 < B < K2 < C < K3 < D
Trick: Identify kink "k1-k3-k2" and lift "k2" twice.
After the first single rotation at k3 (k1's right child) with its left child k2

As you can see, after the first rotation the "inner" problem becomes the "outer" problem, which can be fixed by another rotation.

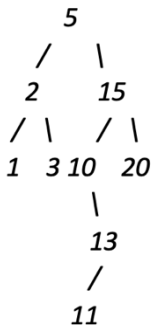
After the second single rotation at k1 with its right child k2



An example of double rotation with the right child
(Give below AVL tree)



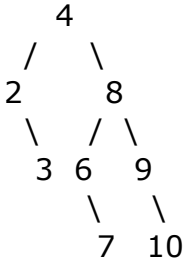
(The result after inserting 11)



Now, we need to perform a double rotation at node ____.
(Draw the result after the first rotation)

(Draw the result after the second rotation)

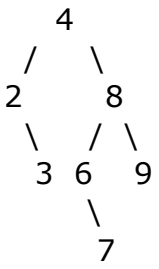
14. Let's consider **deletion** with specific examples.
1) Given the following AVL tree,



deleting 3 will results in: (draw the tree)

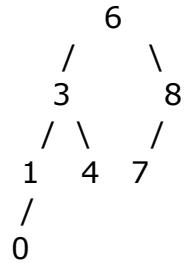
What is the node that we need to rebalance? _____
Since node 8 is balanced, this is essentially the same as rebalancing case ___. Thus we need to perform _____ at node _____.
Draw the result after the rotation.

2) Given the following AVL tree,



deleting 8 will result in (assume successor replacement, not predecessor replacement)
What is the node that we need to rebalance? _____
This is the same as rebalancing of case _____ , thus we need to perform
_____ at node _____ with its _____ child.
Draw the result after the rotation.

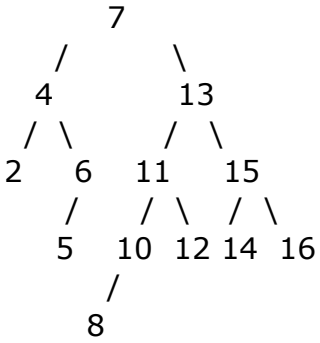
3) Given the following AVL tree,



deleting 7 will result in:

This is the same case as rebalancing case _____, thus we need to perform a _____
_____ at node _____ with its _____ child.
Draw the result after the rotation.

3) Given the following AVL tree,



deleting 4 will result in (assume successor replacement, not predecessor replacement):

What is the node that we need to rebalance?
This is the same as rebalancing of case _____, thus we need to perform _____ _____ at node _____ with its _____ child.
Draw the result after the rotation.

Binary Tree Analysis

0. _____ of a node refers to the distance of the node from the root, measured in the number of e_____.

1. _____ binary tree
a binary tree in which all the levels are completely filled except possibly the lowest (deepest), which is filled from the left.
[Draw some examples]

1) How many nodes at level k (where k is not the lowest level)?
2) Min number of nodes at lowest level?
3) Max number of nodes at lowest level, where k is the height of the tree?
4) Assume, all the levels are completely filled INCLUDING the lowest level
   What is the total number of nodes in this tree with k levels?

5) Given n is the number of nodes in a complete binary tree, what is the number of levels? (Hint: the closest power of 2 that is >= n)

2. _____ binary tree
a binary tree in which every parent node/internal node has either two or no children.
[examples]