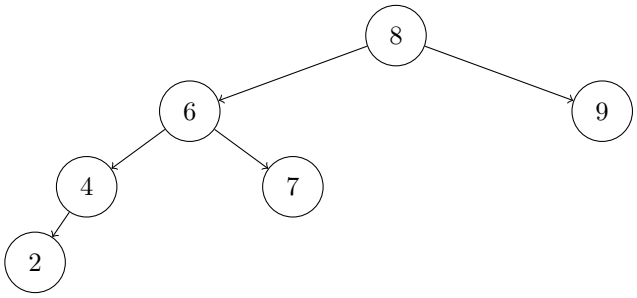
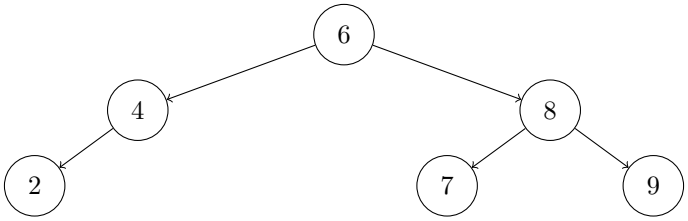
1 AVL Trees

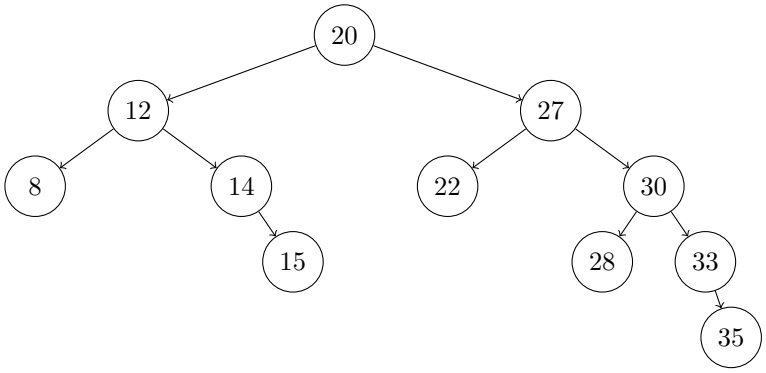
Problem 1. Perform a right rotation on the root of the following tree. Be sure to specify the X, Y, and Z subtrees used in the rotation.



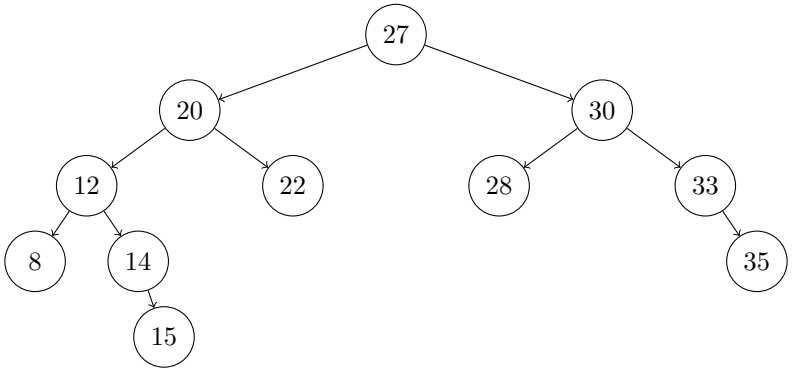
Subtree rooted at 4 is X Subtree rooted at 7 is y Subtree rooted as 9 is Z After the left rotation, the tree will look like



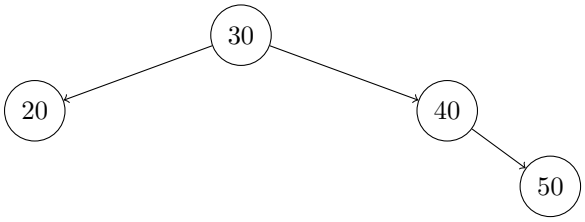
Problem 2. Show the right rotation of the subtree rooted at 27. Be sure to specify the X, Y, and Z subtrees used in the rotation.



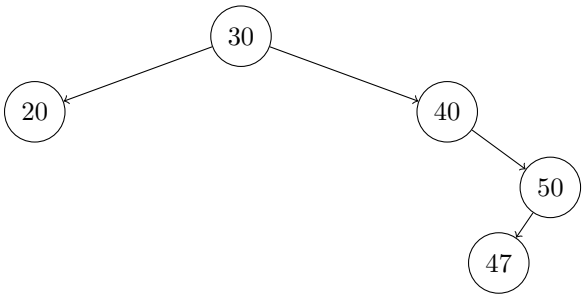
X is the subtree rooted at 12. Y is the subtree rooted at 22, and Z is the subtree rooted at 30. Then, left rotating this,



Problem 3. Using the appropriate AVL tree algorithm, insert the value 47 into the following tree. Show the tree before and after rebalancing.



After inserting 47 (but before rebalancing), the tree will look like this:

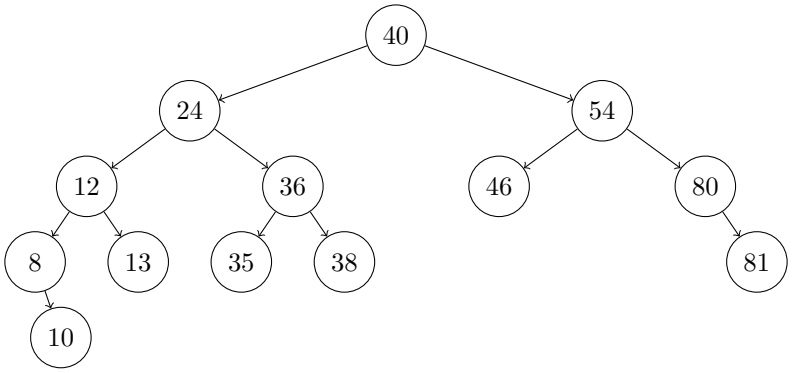


At this point, the tree is out of balance at node ... so we rotate ...

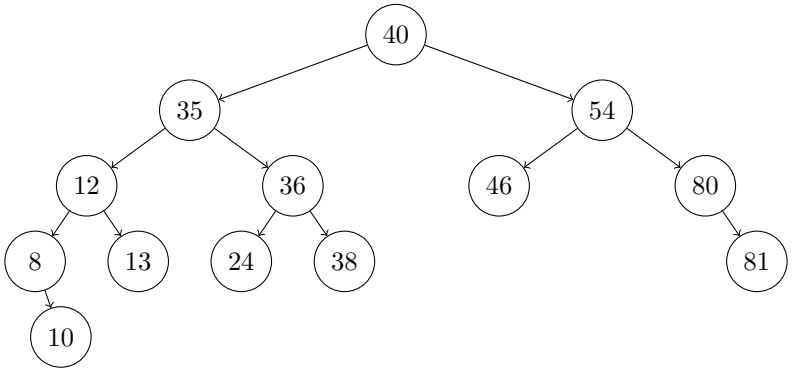
The resulting tree looks like this:

Is the tree now balanced? If not, perform the next rotation here. Repeat until the tree is balanced.

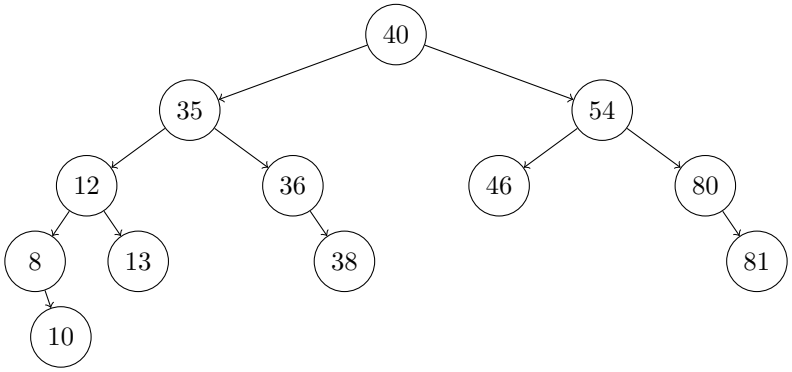
Problem 4. Using the appropriate AVL tree algorithm, remove the value 24 from the following tree. Show the tree before and after *each* rebalancing.



In order to remove 24, we replace 24 with 35 and 35 with 24. Then the tree will look like this.



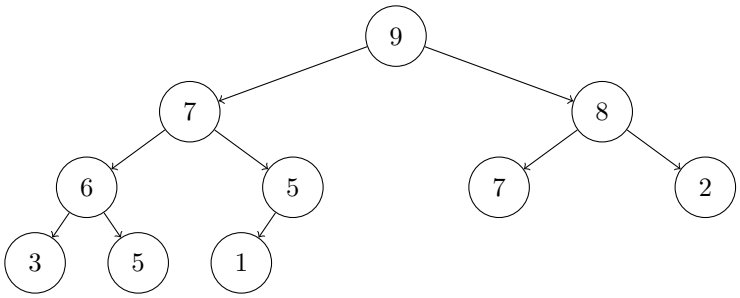
Then we delete the 24. Then the tree will look like this.



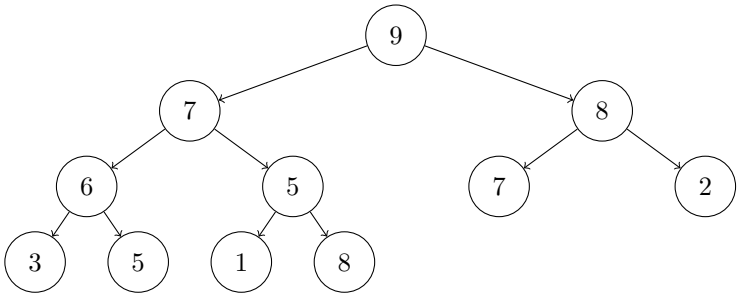
Then, since this tree is already balanced, we don't need to do anything further.

2 Heaps

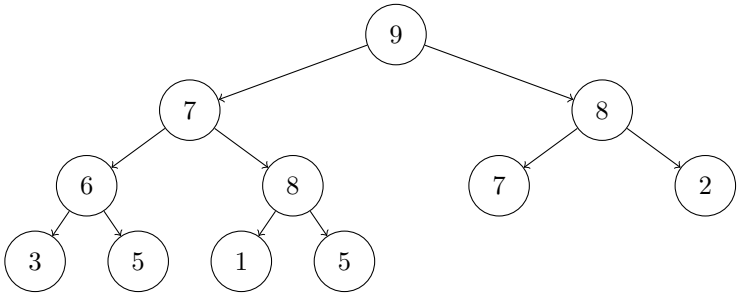
Problem 1. Show the addition of the element 8 to the max-heap below. First, show the addition of 8 to the tree; then, show each bubbling step.



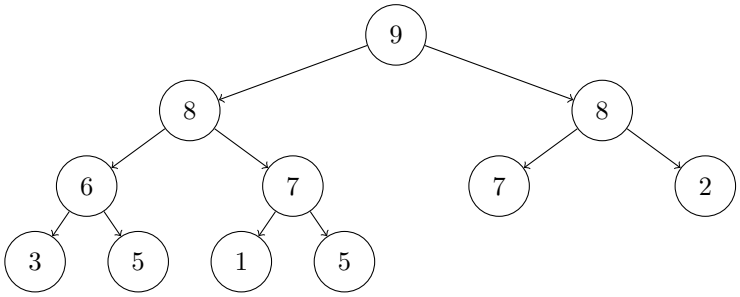
The value 8 is initially added as follows:



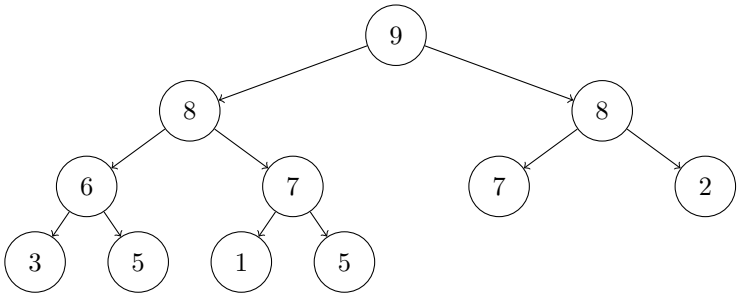
We swap 8 with 5 as our first swap:



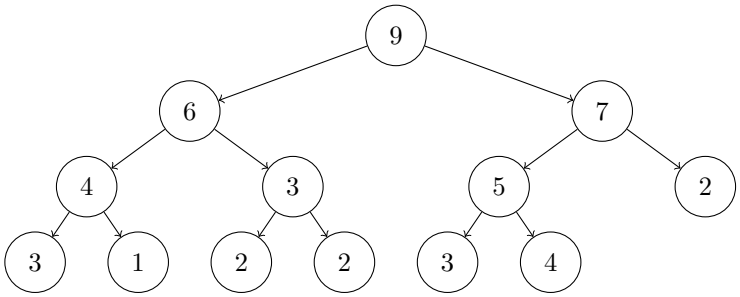
Now, we swap 7 with 8 as our first swap. :



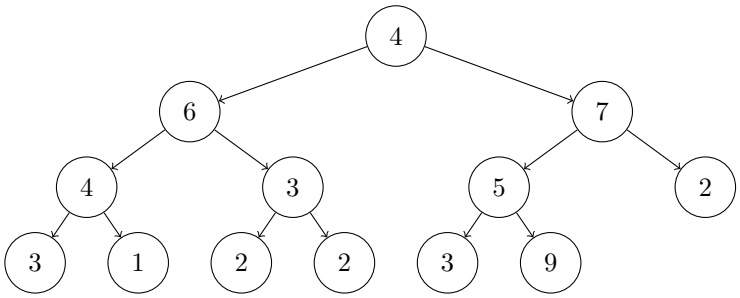
We cannot swap 8 with 9 since 9 is larger than 8. Hence, our tree looks like:



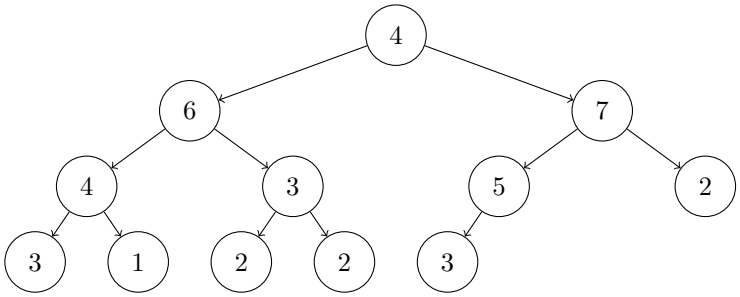
Problem 2. Show the removal of the top element of this max-heap. First, show the swap of the root node; then, show each bubbling step.



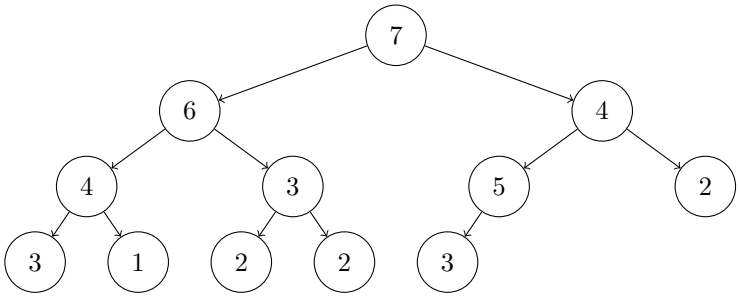
We first swap 9 with the last node in the last row which is 4:



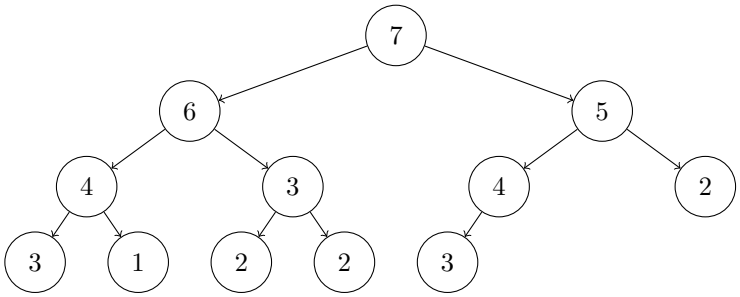
We delete the last node now and are left with the following :



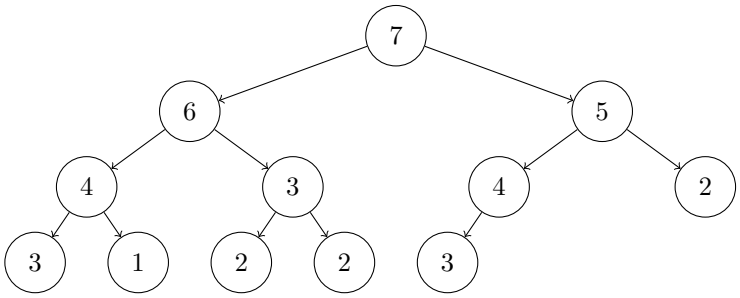
We now start bubbling down the root of the tree. We first swap 4 with 7



We now swap 4 with 5 to get the following tree:

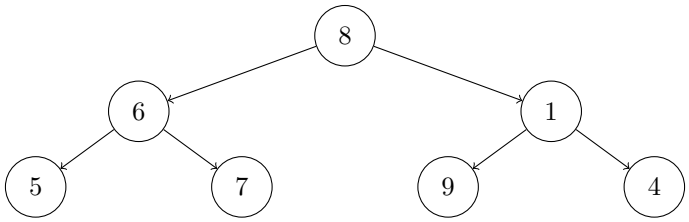


We cannot do any further bubbling down and so our final tree is :

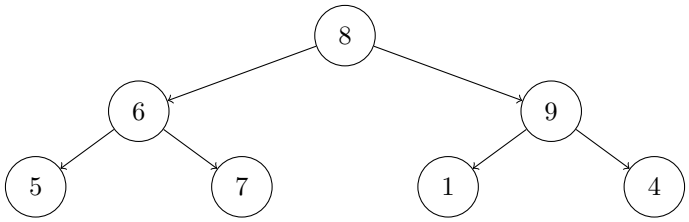


Problem 3. Consider the sequence of elements `[8,6,1,5,7,9,4]`. Using the representation discussed in class, show the tree to which this sequence corresponds. Then, show the *heapification* of this tree; that is, show how this tree is transformed into a heap. Demonstrate each bubbling step.

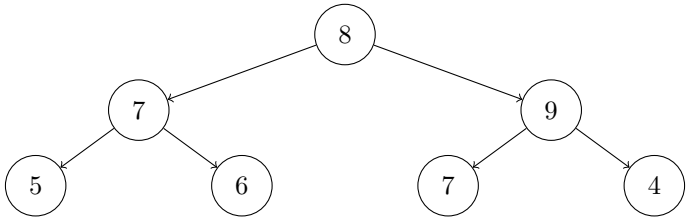
Here is the list interpreted as a complete binary tree



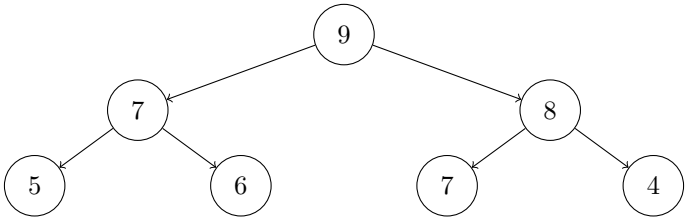
Doing the first bubble-down in the right-subtree of root 8,



Doing the second bubble-down in the left-subtree of root 8,



Then, with the final bubble-down step regarding the root 8 and its two children,


9
7
8
5
6
7
4