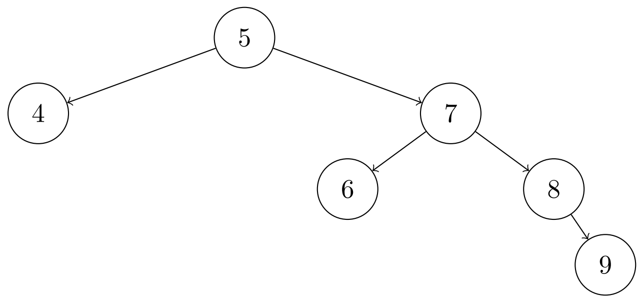
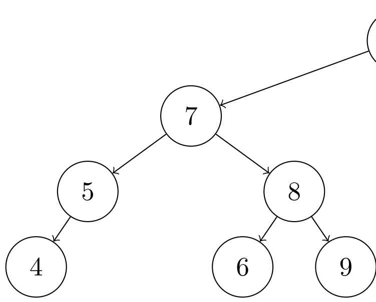
1 AVL Trees

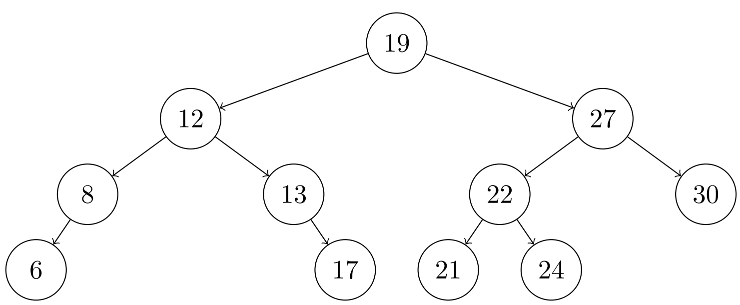
Problem 1. Perform a left rotation on the root of the following tree. Be sure to specify the X, Y, and Z subtrees used in the rotation.



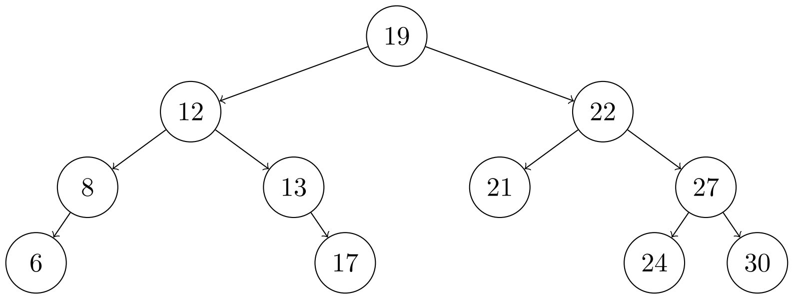
X Subtree: 4 Y Subtree: 6 Z Subtree: 8



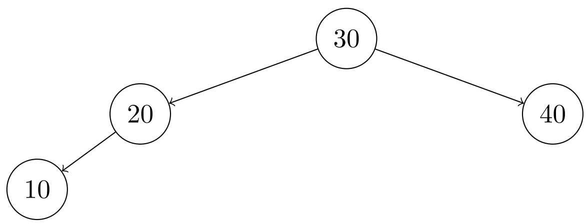
Problem 2. Show the right rotation of the subtree rooted at 27. Be sure to specify the X, Y, and Z subtrees used in the rotation.



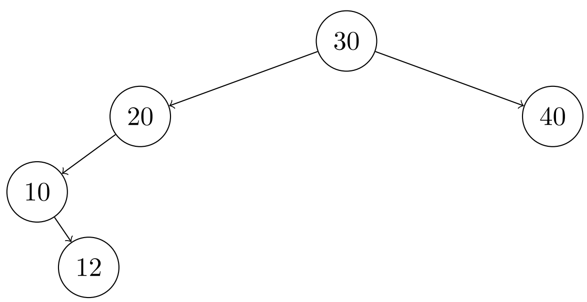
X Subtree: 21 Y Subtree: 24 Z Subtree: 30



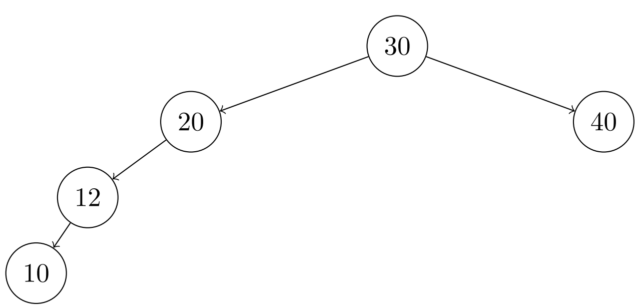
Problem 3. Using the appropriate AVL tree algorithm, insert the value 12 into the following tree. Show the tree before and after rebalancing.



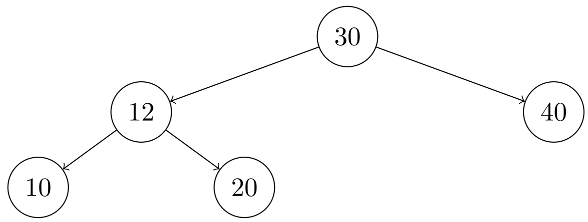
Since 12 is smaller than the root, 30, it goes to the left, and since 12 is smaller than 20, it goes to the left Since 12 is larger than 10, it goes to the right



The tree is unbalanced since the LSTH and RSTH of 20 (2) - (0) greater than 1 Since the Y subtree height is greater than the x subtree height, a left rotation is performed around 12

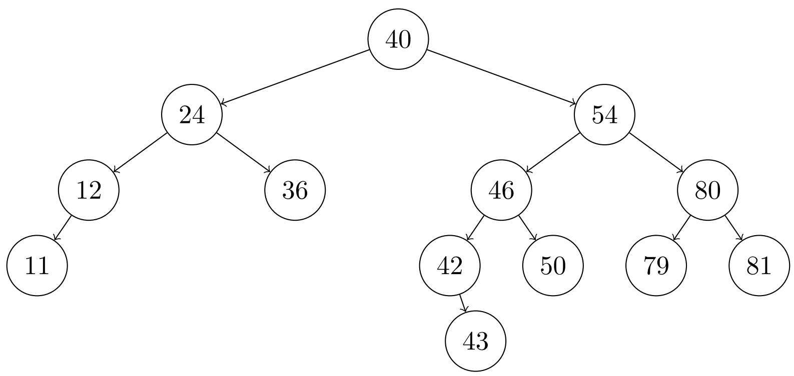


Now a right rotation is performed around 20

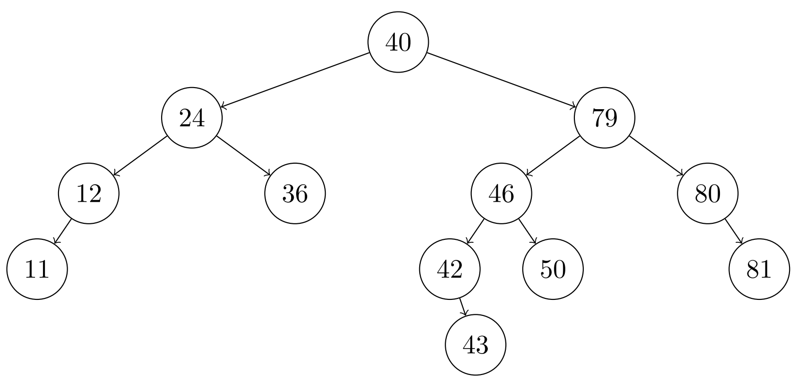


The tree is now balanced

Problem 4. Using the appropriate AVL tree algorithm, remove the value 54 from the following tree. Show the tree before and after rebalancing.



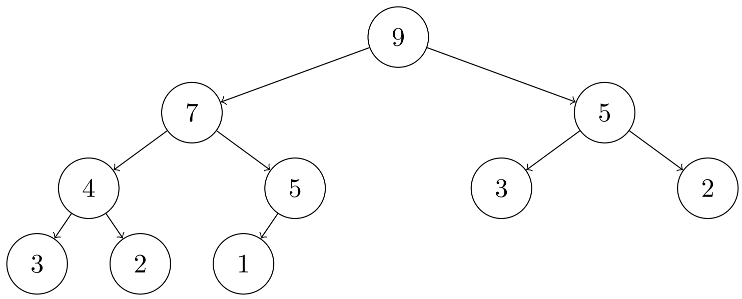
Since 79 is the smallest value of the right most subtree of 54, 79 is swapped with 54, and 54 is removed



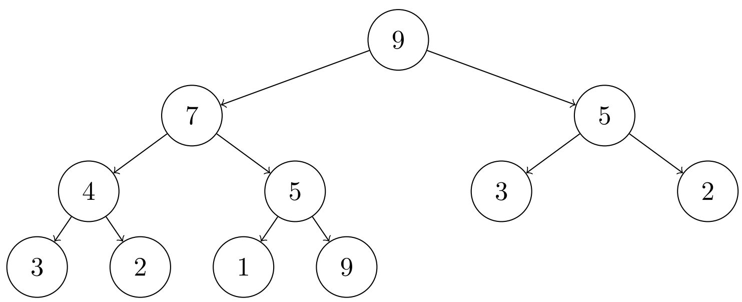
The tree is now balanced.

2 Heaps

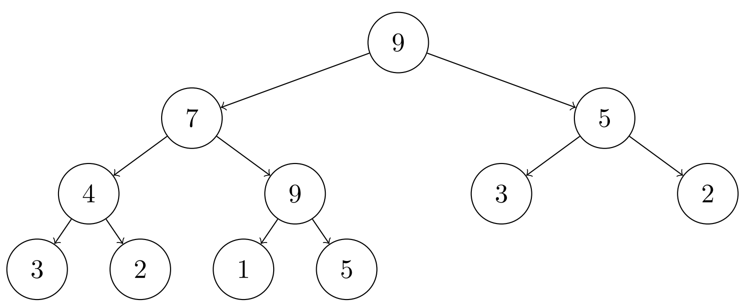
Problem 1. Show the addition of the element 9 to the max-heap below. First, show the addition of 9 to the tree; then, show each bubbling step.



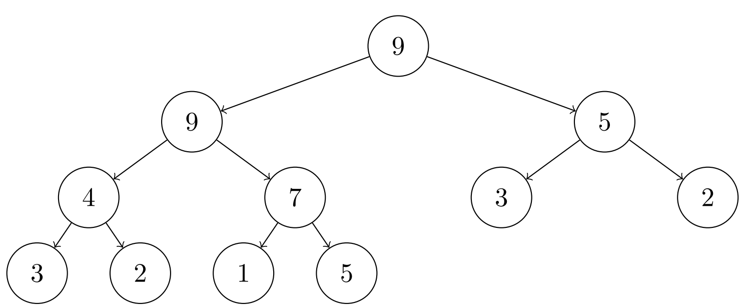
Since the BT has to remain complete, 9 will be inserted as the right child of the left 5.



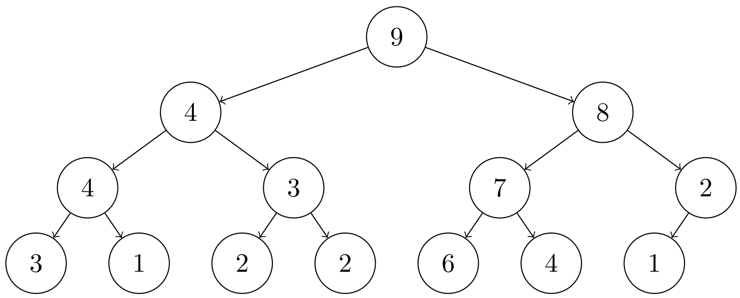
The max-heap invariant must be met, so 9 is swapped with 5 since 9 greater than 5.



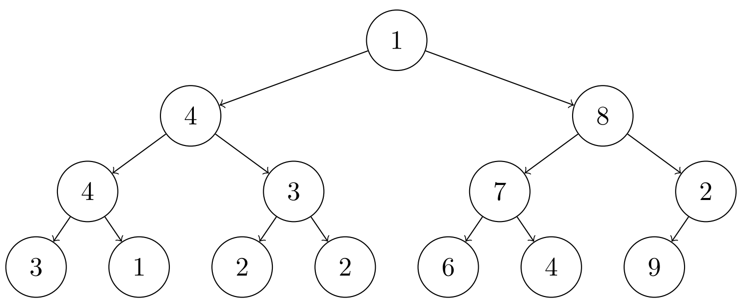
Since 9 is now the child of 7, it must also swap. This leaves 9 to be a child of the root 9, which meets the heap invariant.



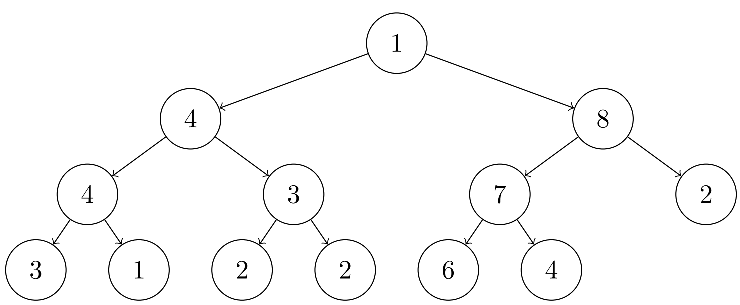
Problem 2. Show the removal of the top element of this max-heap. First, show the swap of the root node; then, show each bubbling step.



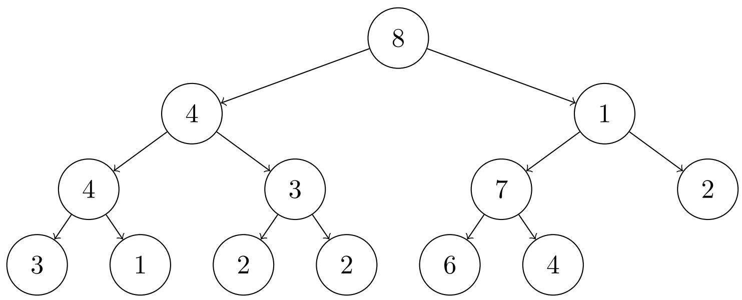
Since 9, the root, is being removed, it will be swapped with last node, which is 1.



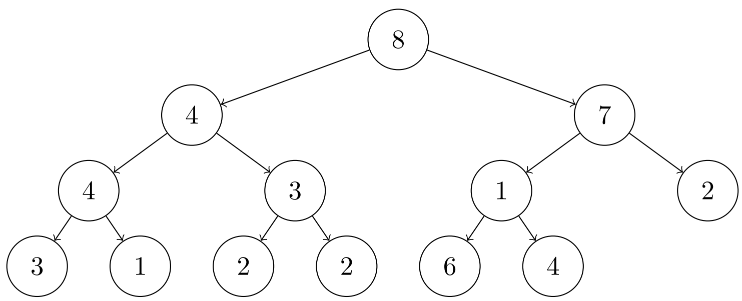
Once swapped, 9 can be removed. 1 is now the root.



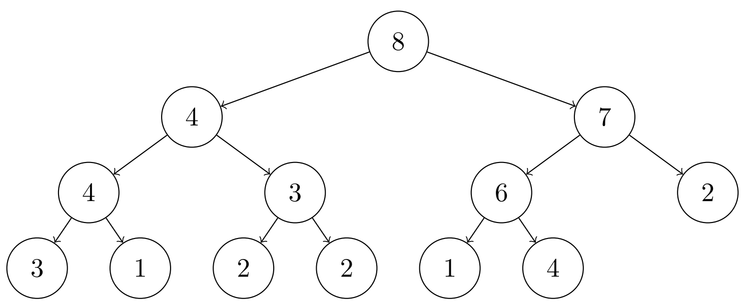
To meet the heap invariant, 1 will be swapped with either its right or left child. Since rightChild(8) greater than leftChild(4), 1 is swapped with 8.



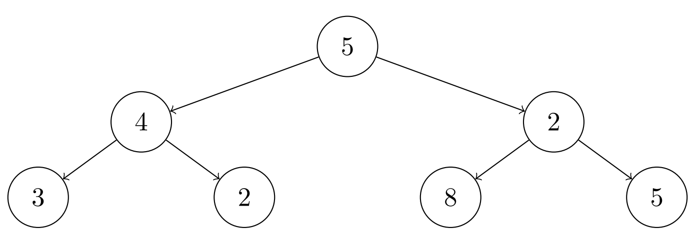
Now that 1 has been bubbledDown, it's compared with its new children. In this case, it's swapped with 7 since 7 greater than 2.



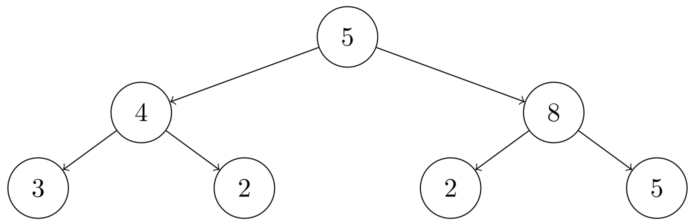
Like before, 1 is swapped. This time it is with 6 since 6 greater than 4. 1 becomes a leafNode, so it stops there.



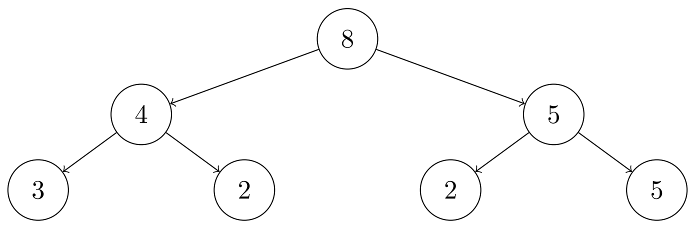
Problem 3. Consider the sequence of elements `[5,4,2,3,2,8,5]`. Using the representation discussed in class, show the tree to which this sequence corresponds. Then, show the *heapification* of this tree; that is, show how this tree is transformed into a heap. Demonstrate each bubbling step.



The 8 would first swap with 2, to make the right Subtree a heap. Since the left Subtree is already a heap, no swapping is needed.



Since both subtrees are heaps, the root 5 can now be bubbled down. 8 greater than 4, so 5 is swapped with 8



With having met the max-heap invariant, the Complete Binary Tree is now a heap.