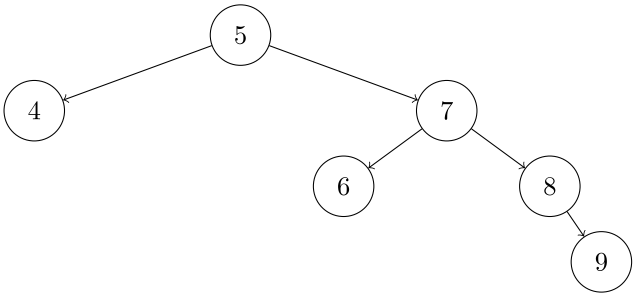
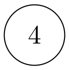
1 AVL Trees

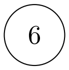
Problem 1. Perform a left rotation on the root of the following tree. Be sure to specify the X, Y, and Z subtrees used in the rotation.



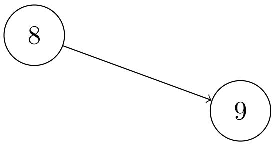
X subtree



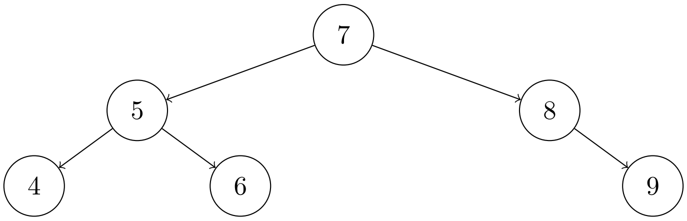
Y subtree



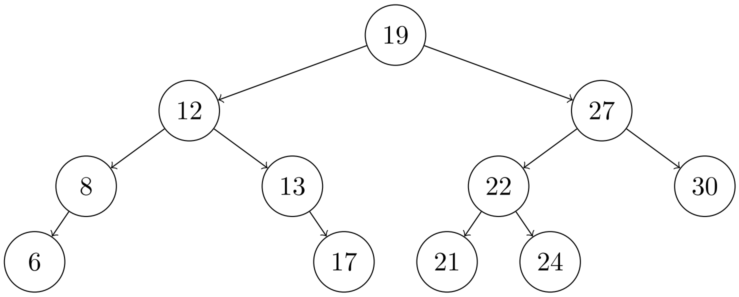
Z subtree



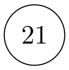
after rotation



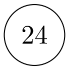
Problem 2. Show the right rotation of the subtree rooted at 27. Be sure to specify the X, Y, and Z subtrees used in the rotation.



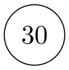
X subtree



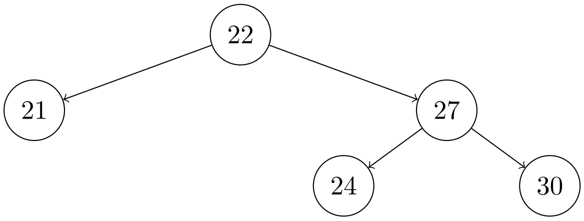
Y subtree



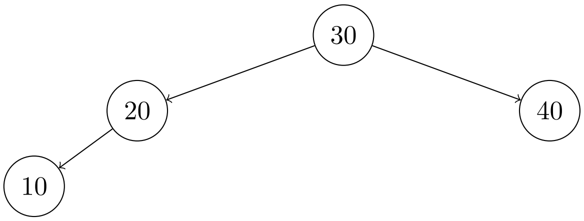
Z subtree



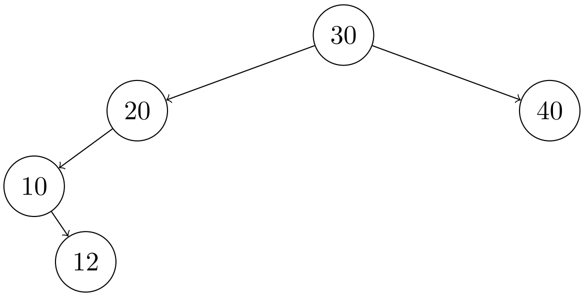
after rotation



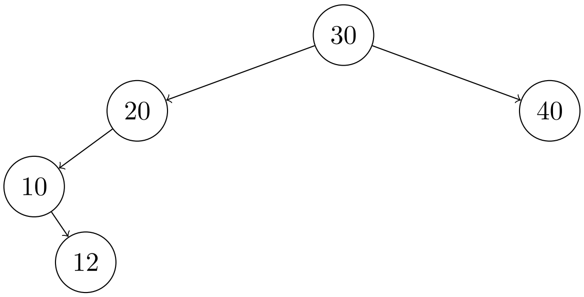
Problem 3. Using the appropriate AVL tree algorithm, insert the value 12 into the following tree. Show the tree before and after rebalancing.



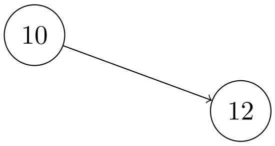
insert(12)



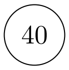
Before rebalance



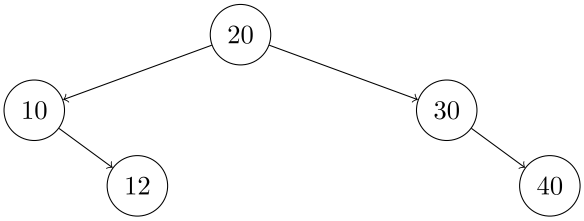
X subtree



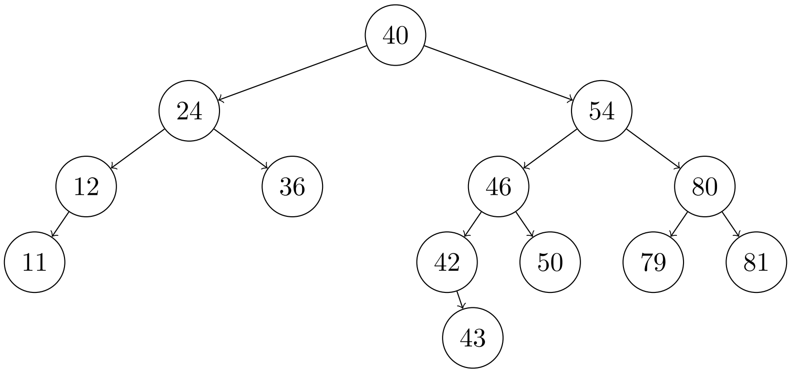
Y subtree (empty) Z subtree



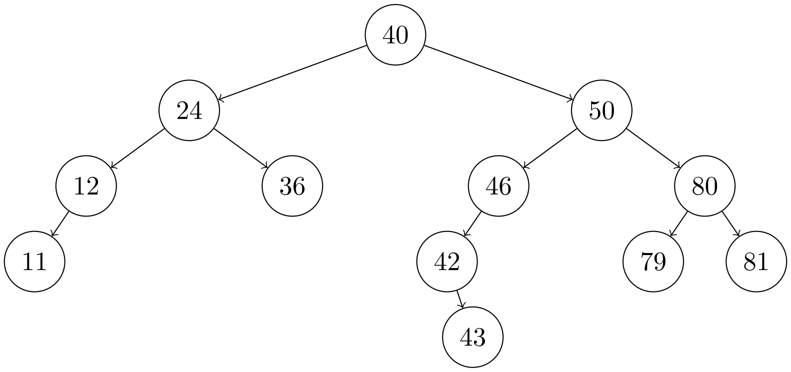
Rebalanced



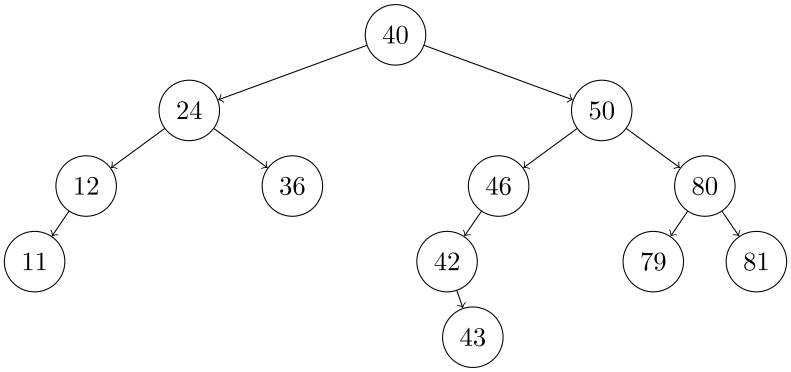
Problem 4. Using the appropriate AVL tree algorithm, remove the value 54 from the following tree. Show the tree before and after rebalancing.



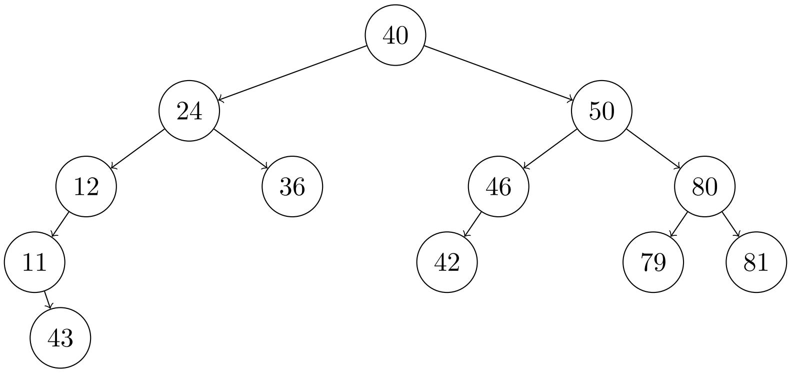
remove(54)



before rebalanced

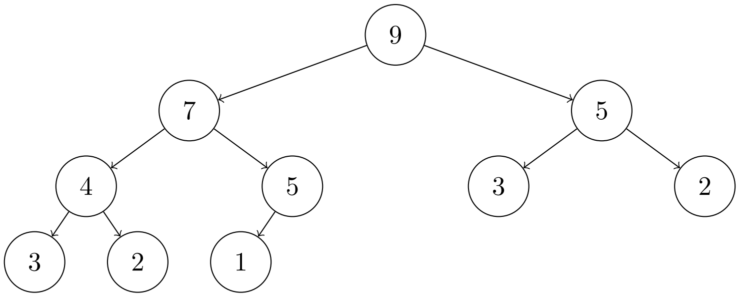


no balancing is needed, still a valid AVL tree

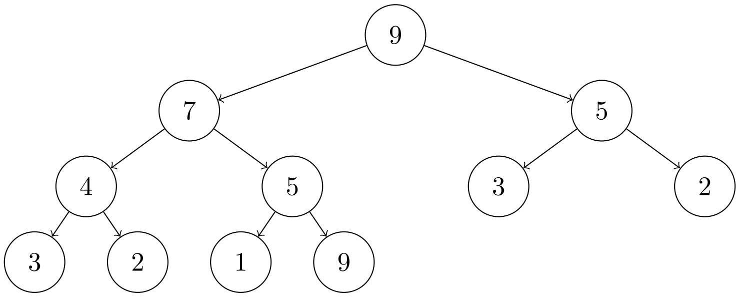


2 Heaps

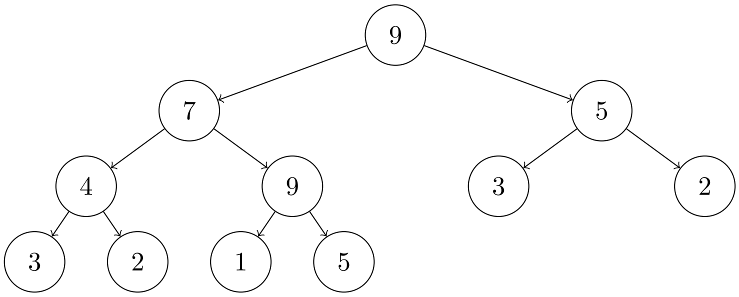
Problem 1. Show the addition of the element 9 to the max-heap below. First, show the addition of 9 to the tree; then, show each bubbling step.



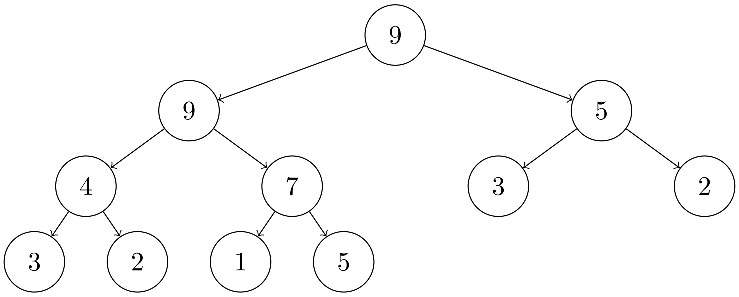
insert(9)



9 > 5? yes => bubbleUp

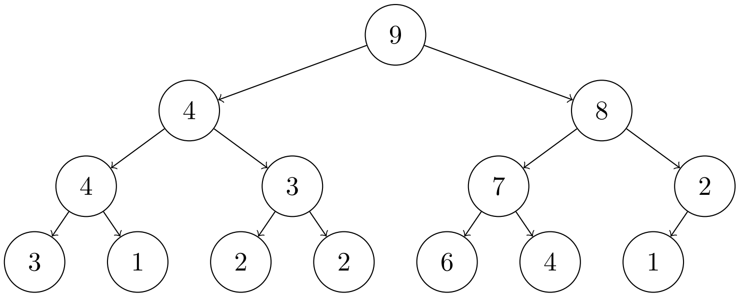


9 > 7? yes => bubbleUp

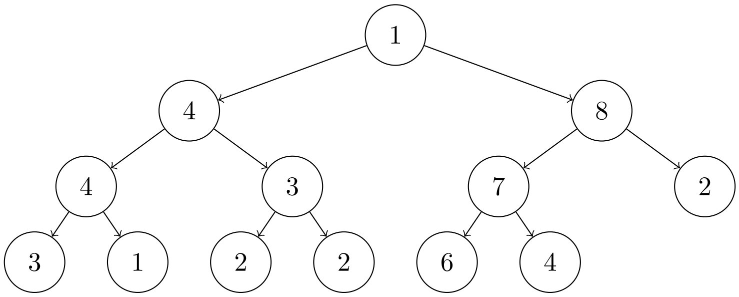


9 > 9 ? no => done

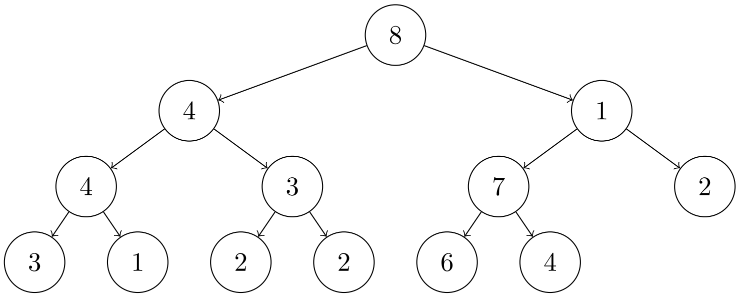
Problem 2. Show the removal of the top element of this max-heap. First, show the swap of the root node; then, show each bubbling step.



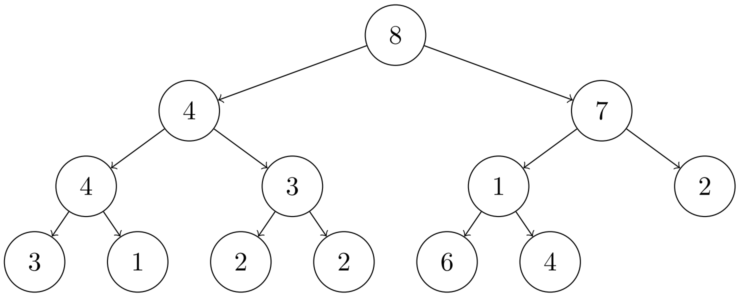
remove()



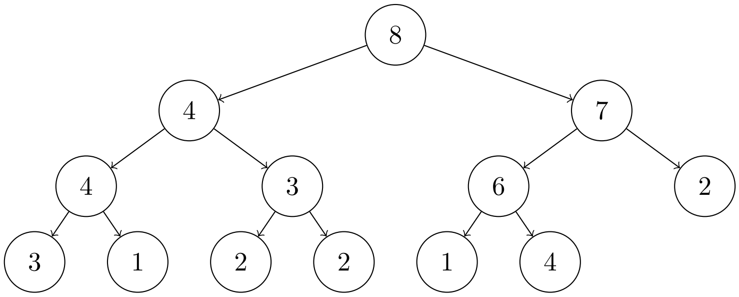
1 < max(4, 8)? yes => swap(1, 8)



1 < max(7, 2)? yes => swap(1, 7)



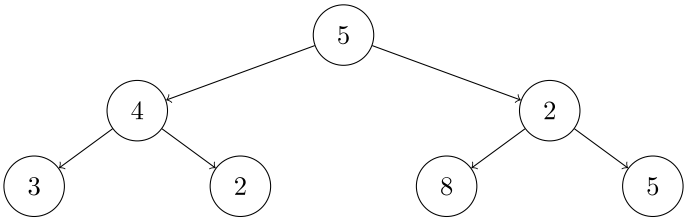
1 < max(6, 4)? yes => swap(1, 6)



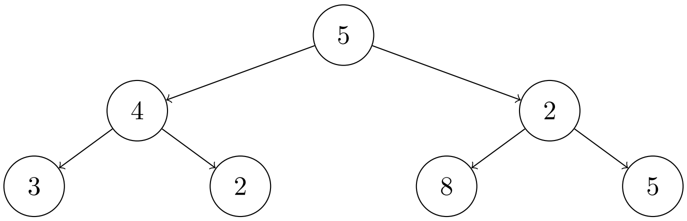
1 is now, so done

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.

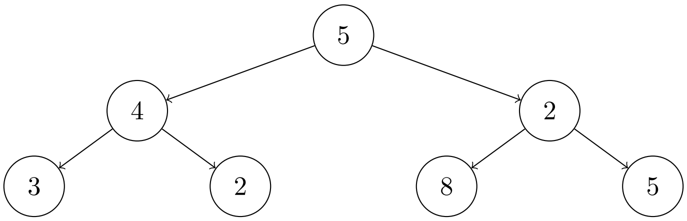
Convert [5,4,2,3,2,8,5] to complete binary tree



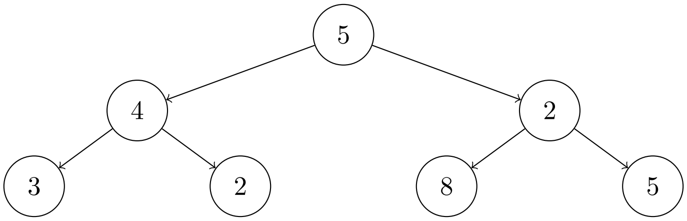
i = 0, node at i = 5, bubbleUp(i)



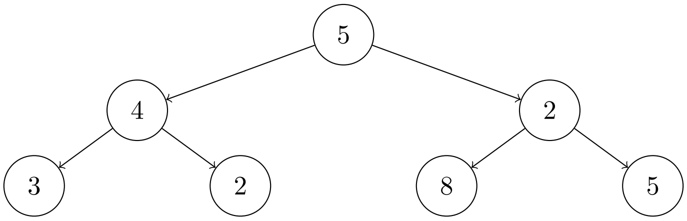
i = 1, node at i = 4, bubbleUp(i)



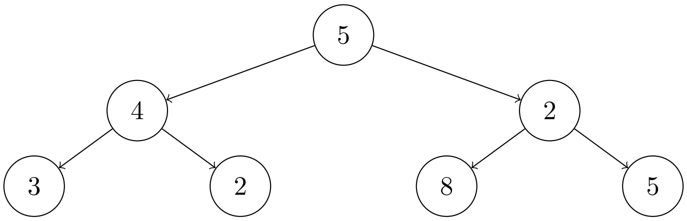
i = 2, node at i = 2, bubbleUp(i)



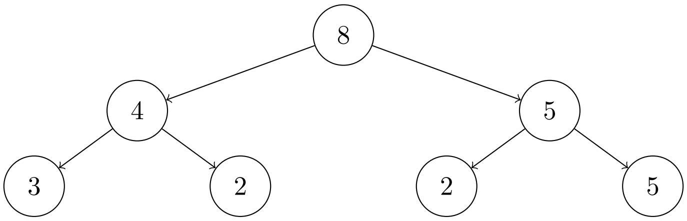
i = 3, node at i = 3, bubbleUp(i)



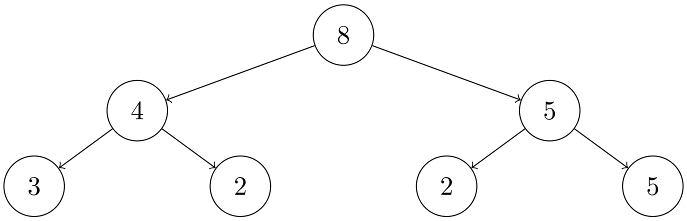
i = 4, node at i = 2, bubbleUp(i)



i = 5, node at i = 8, bubbleUp(i)



i = 6, node at i = 5, bubbleUp(i)



i = size-1 => loop ends