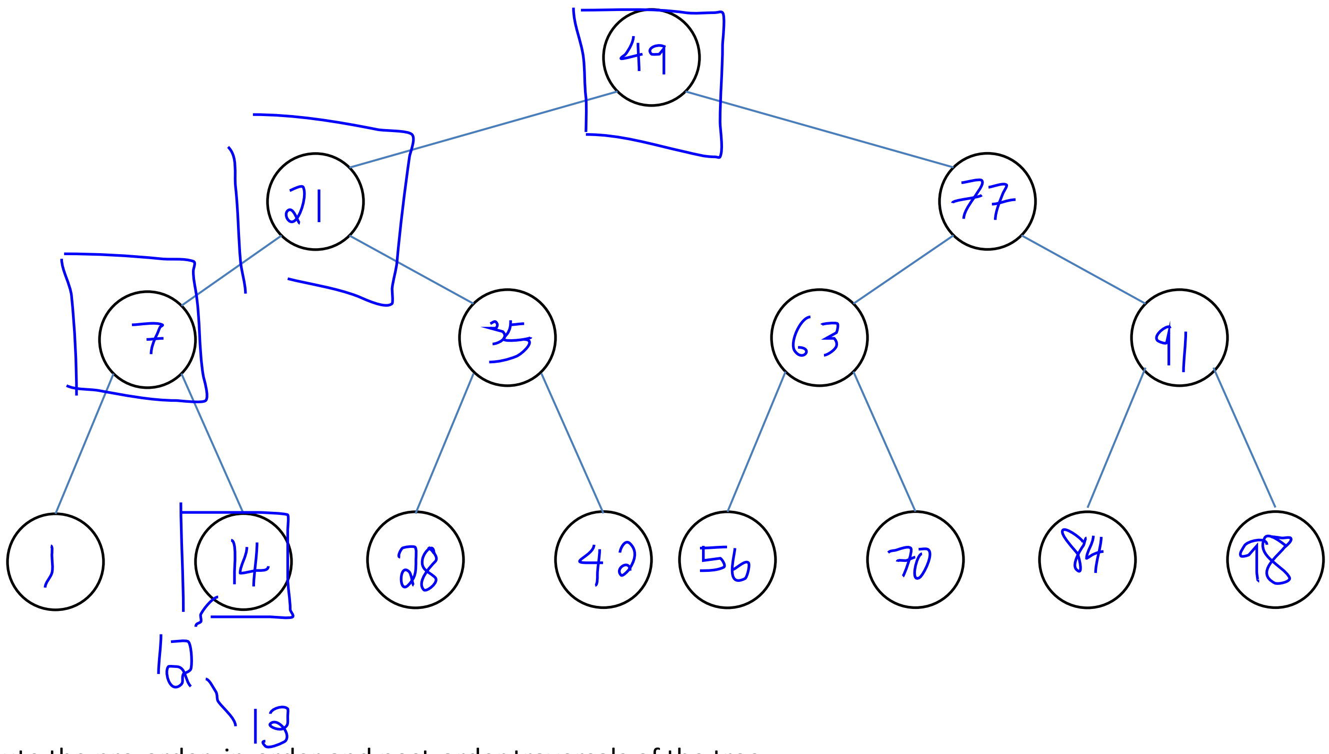
CS2040C Data Structure and Algorithm
AVL Tree Exercise

(Hint: You can do this exercise together with Visualgo for animation.)

Given an empty tree, insert the following integers by the following order. You do not have to worry about balancing for now.

49, 21, 7, 1, 14, 35, 28, 42, 77, 63, 56, 70, 91, 84, 98



Compute the pre-order, in-order and post-order traversals of the tree

Pre-order traversal:

49, 21, 7, 1, 14, 35, 28, 42, 77, 63, 56, 70, 91, 84, 98

In-order traversal:

1, 7, 14, 21, 28, 35, 43, 49, 56, 63, 70, 77, 84, 91, 98

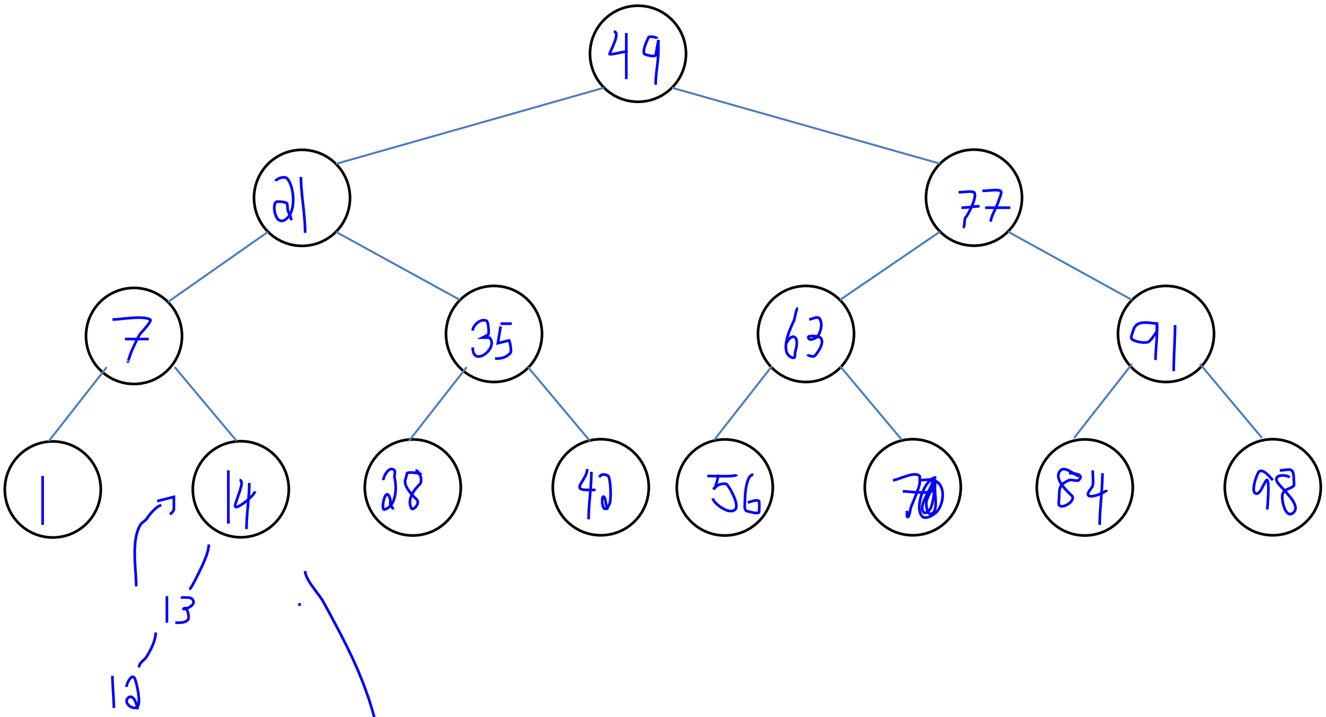
Post-order traversal

1, 14, 7, 28, 42, 35, 21, 56, 70, 63, 84, 98, 91, 77, 49

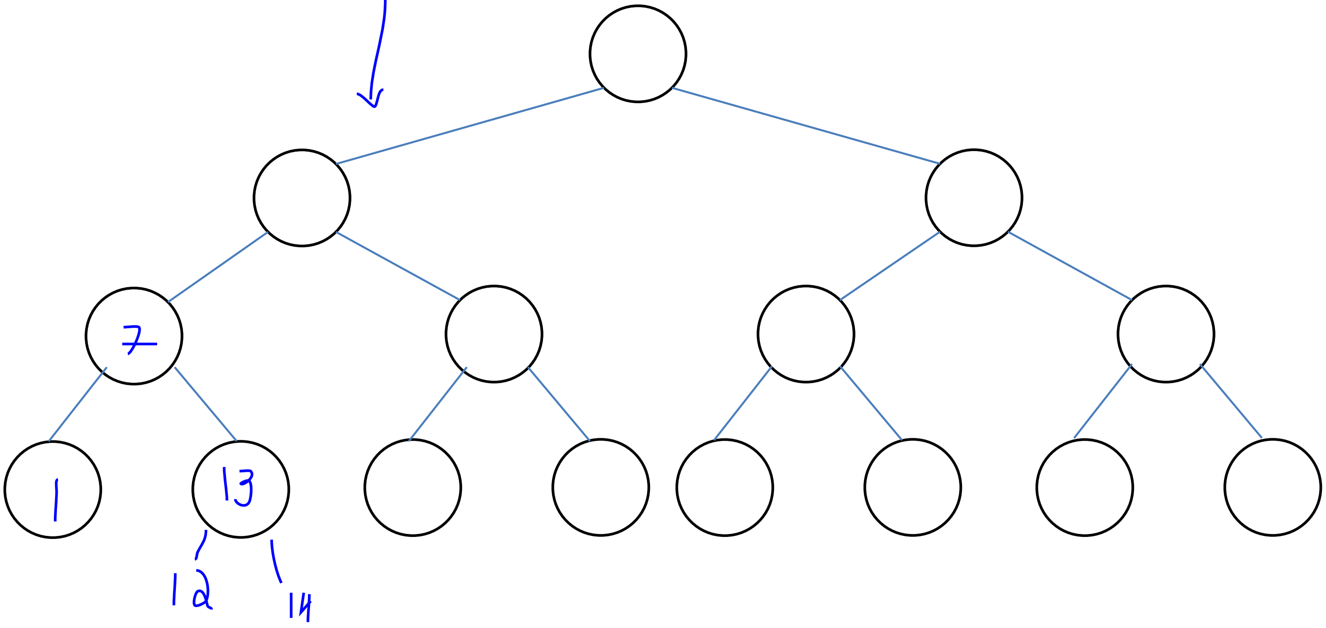
In the previous diagram, insert 12, then insert 13. Draw more nodes If you need to. Circle ALL nodes that are not height-balanced according to AVL tree definition.

Which one is the lowest one (closest to the newly inserted node 13) that is not height balanced? Is it left or right heavy?

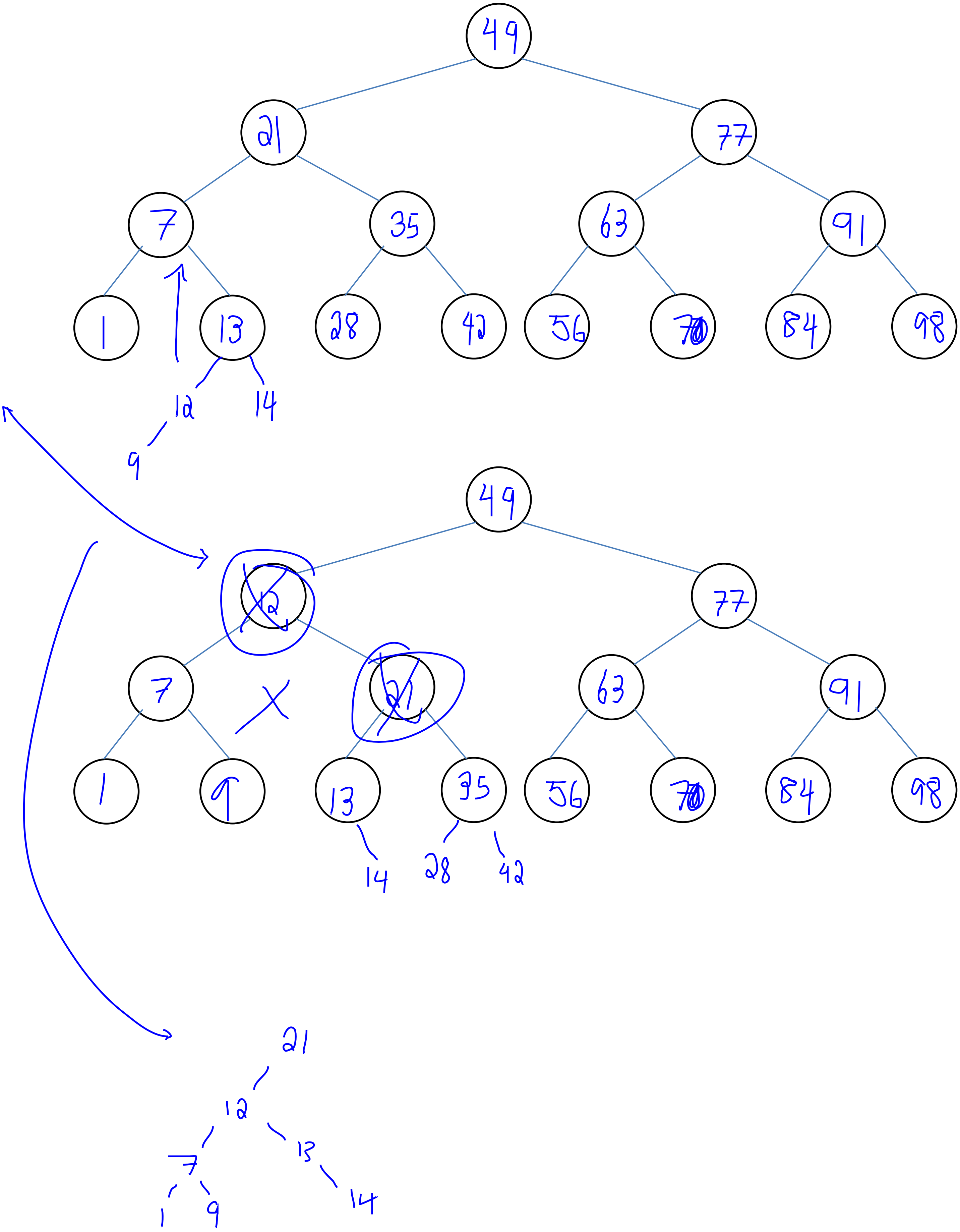
According to the notes, how many rotation(s) should we do? Draw the tree after ONE rotation



Then the next tree after the second rotation



Try to rebalance the tree after insert the number 9


49
21
77
7
35
63
91
1
13
28
42
56
70
84
98
12
14
9
49
12
77
7
21
63
91
1
9
13
35
56
70
84
98
14
28
42
21
12
7
13
1
9
14