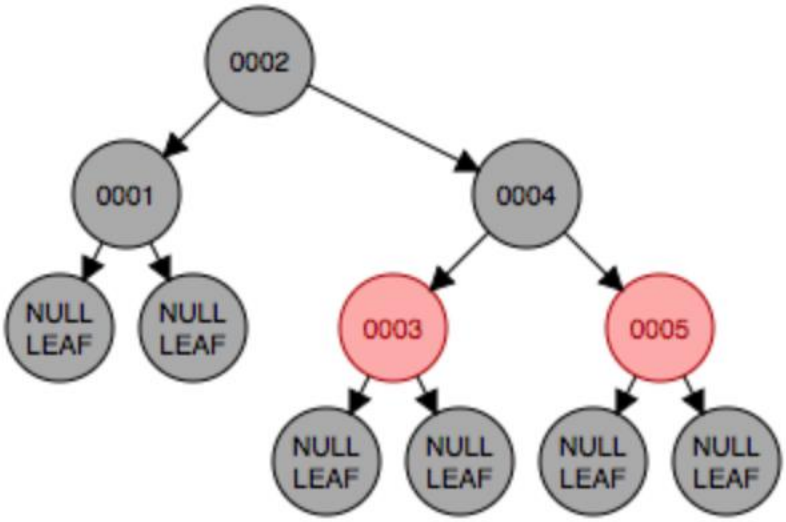
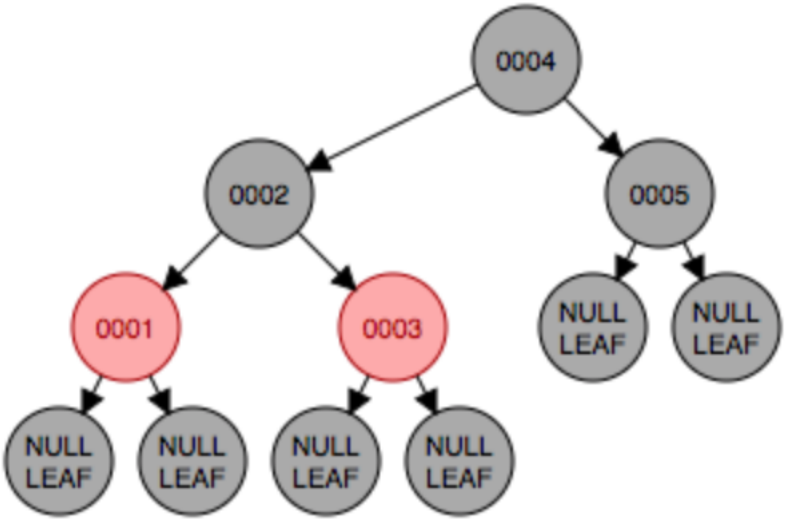
Detailed solution for 1A

There are 6 different variation to store 1 to 5 depending on the order of insertion.

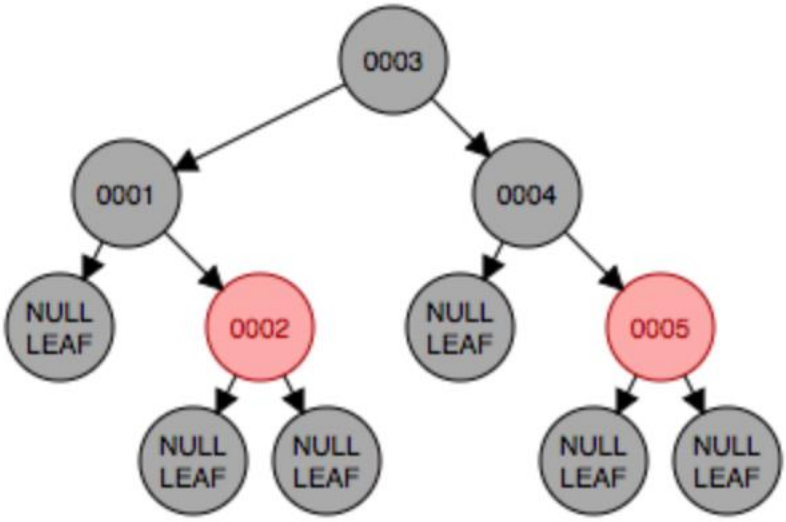
1)



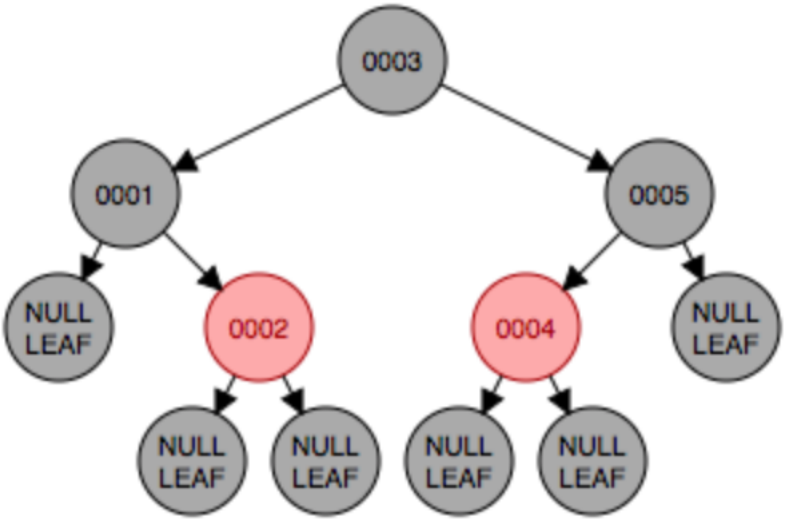
2)



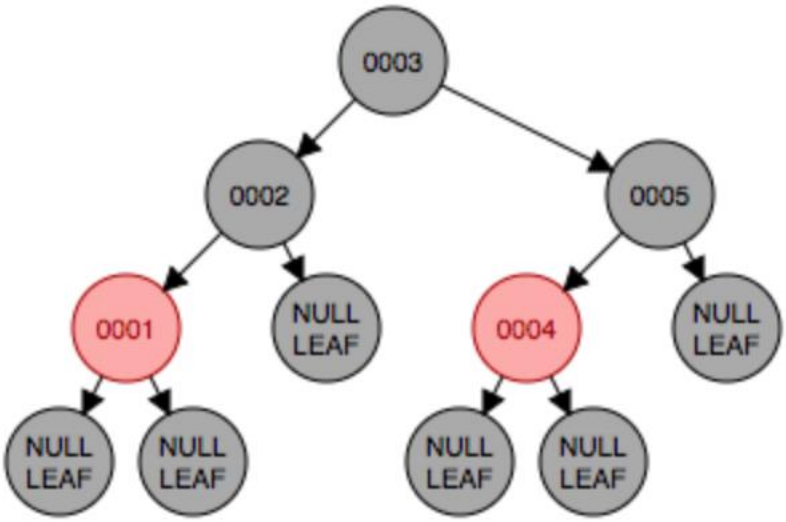
3)


0003
0001
0004
NULL LEAF
0002
NULL LEAF
0005
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF

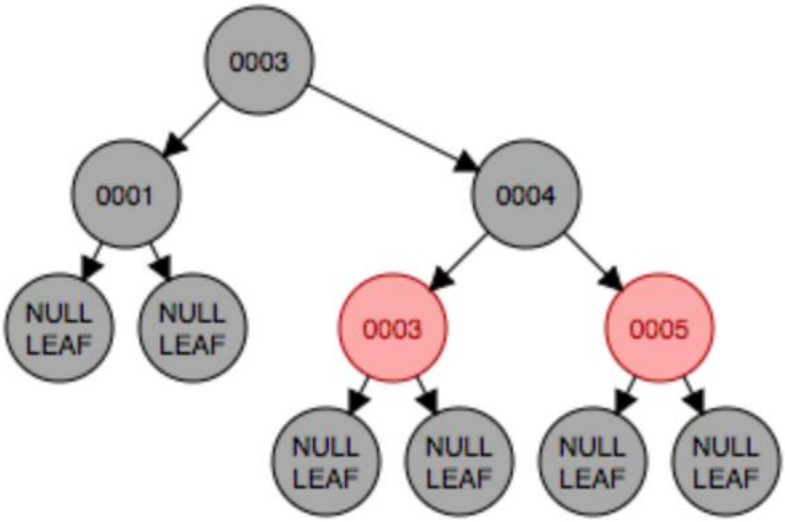
4)


0003
0001
0005
NULL LEAF
0002
0004
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF

5)

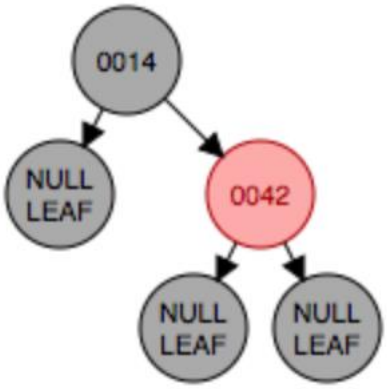

0003
0002
0005
0001
NULL LEAF
0004
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF

6)

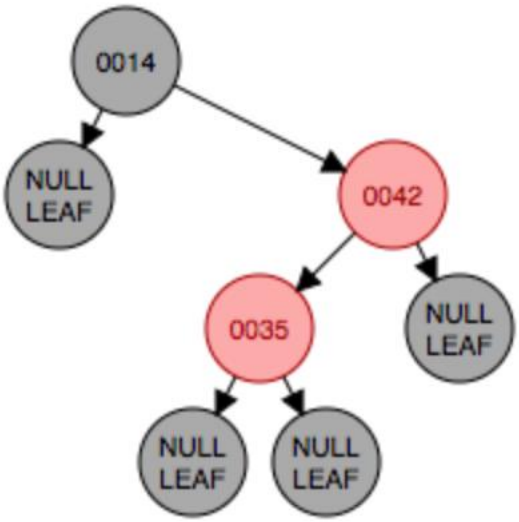

0003
0001
0004
NULL LEAF
NULL LEAF
0003
0005
NULL LEAF
NULL LEAF
NULL LEAF
NULL LEAF

Detailed solution for 1B:
The following is the detailed steps for what happens when you insert all of those values in that specific order.
14 is first element and it's a black node.

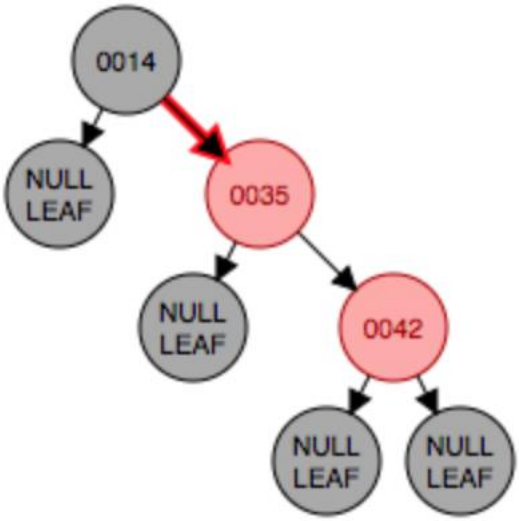
42 is added and its larger than 14 so it goes to the right and is inserted as a red node.



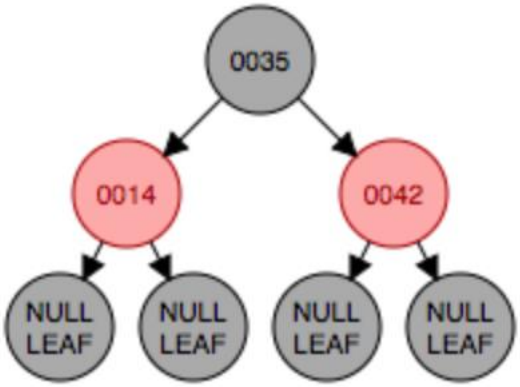
35 is inserted and is found to be larger than 14 and smaller than 42. But now 2 red nodes succeed one another. For that reason action must be taken.



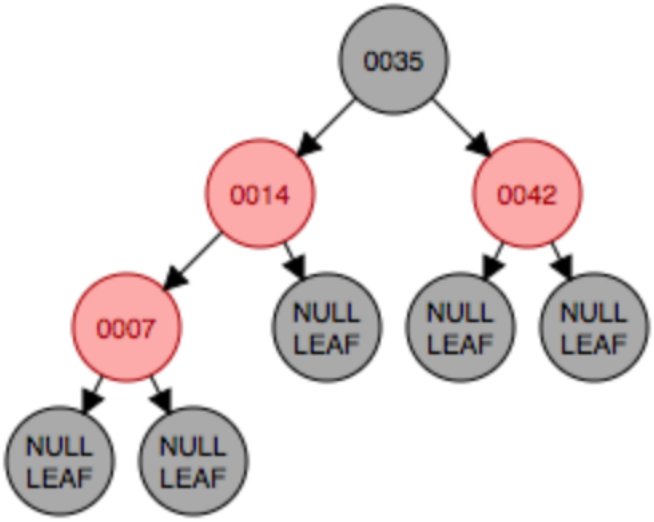
Rotation is called but it is still 2 successive red nodes



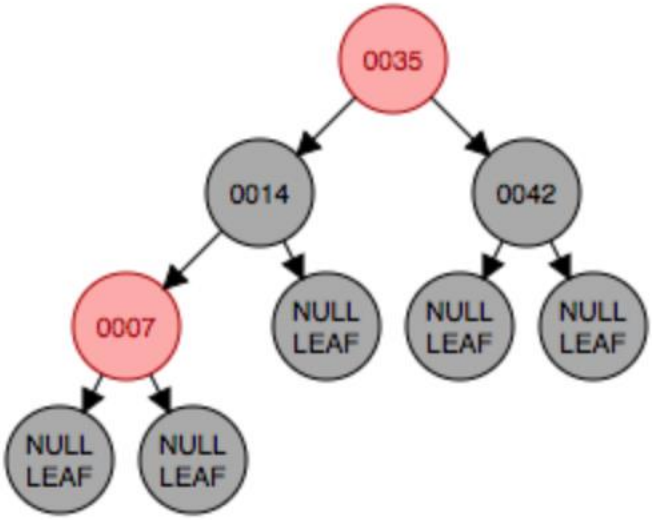
Another rotation is called then 14 and 35 change colors



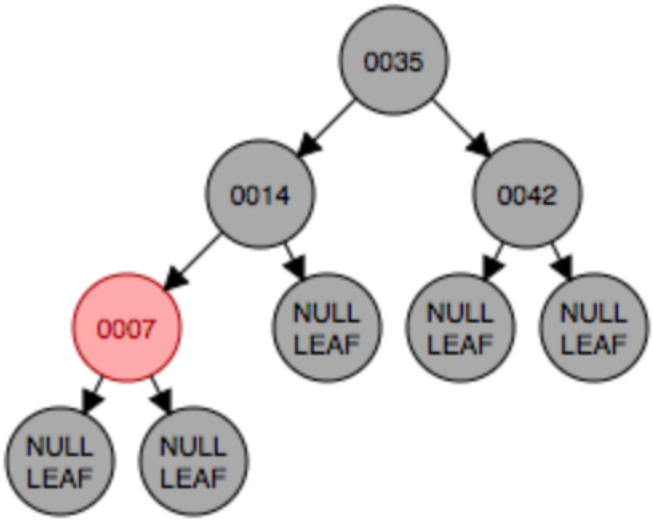
7 is added but then we have 2 successive red nodes.



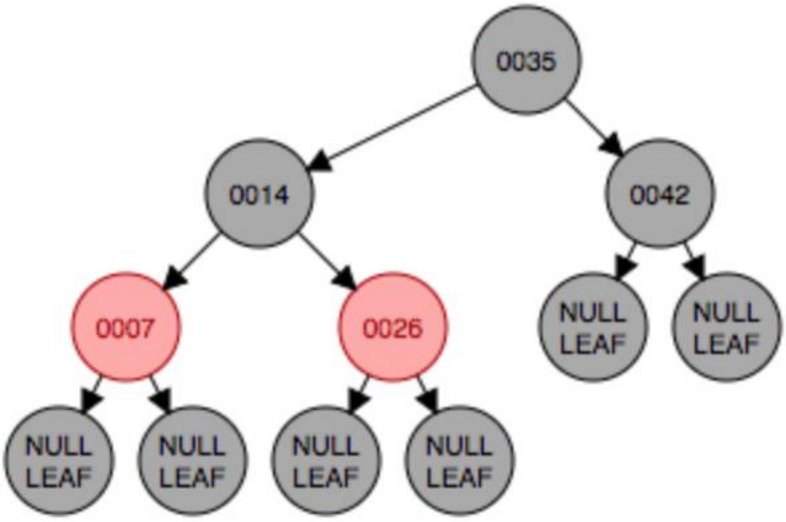
We then change the colors of the top 3 nodes in order to create a balanced 3 with correct black height



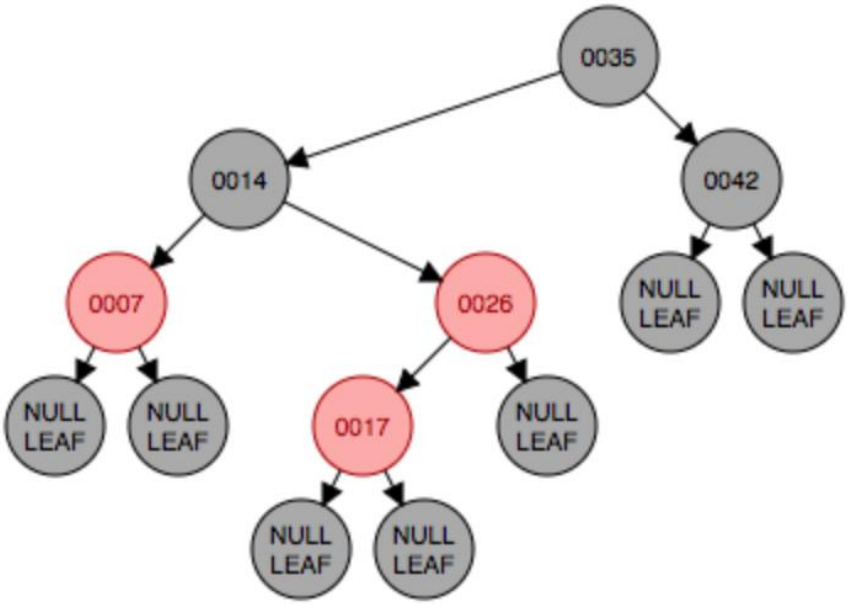
We then change 35 to black because root must be black.



26 inserted to the tree. No errors



17 inserted but now we have two successive red nodes.



Change the colors of the 14 and 26 node and then it is balanced with the correct black height