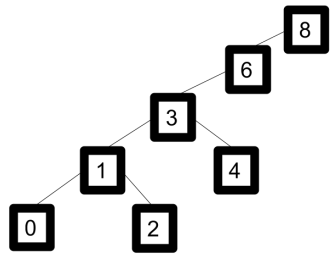
Balancing Trees

Here is a video walkthrough of the solutions.

We are given the following extremely unbalanced search tree.



Select the minimum number of rotations in the correct order required to balance this tree. *Hint*: The resulting tree should have two layers of nodes below the root.

[ ] Rotate left on 8
[ ] Rotate right on 8
[ ] Rotate left on 6
[ ] Rotate right on 6
[ ] Rotate left on 4
[ ] Rotate right on 4
[ ] Rotate left on 3
[ ] Rotate right on 3
[ ] Rotate left on 2
[ ] Rotate right on 2
[ ] Rotate left on 1
[ ] Rotate right on 1
[ ] Rotate left on 0
[ ] Rotate right on 0

Solution:
[ ] Rotate left on 8
[X] Rotate right on 8
[ ] Rotate left on 6
[X] Rotate right on 6
[ ] Rotate left on 4
[ ] Rotate right on 4
[ ] Rotate left on 3
[ ] Rotate right on 3
[ ] Rotate left on 2
[ ] Rotate right on 2
[ ] Rotate left on 1
[ ] Rotate right on 1
[ ] Rotate left on 0
[ ] Rotate right on 0

Explanation: Rotating right on 8, then on 6, makes 3 the new root of the tree (with 6 as the right child). Verify this for yourself.