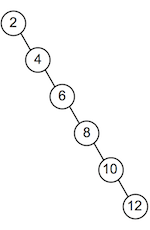
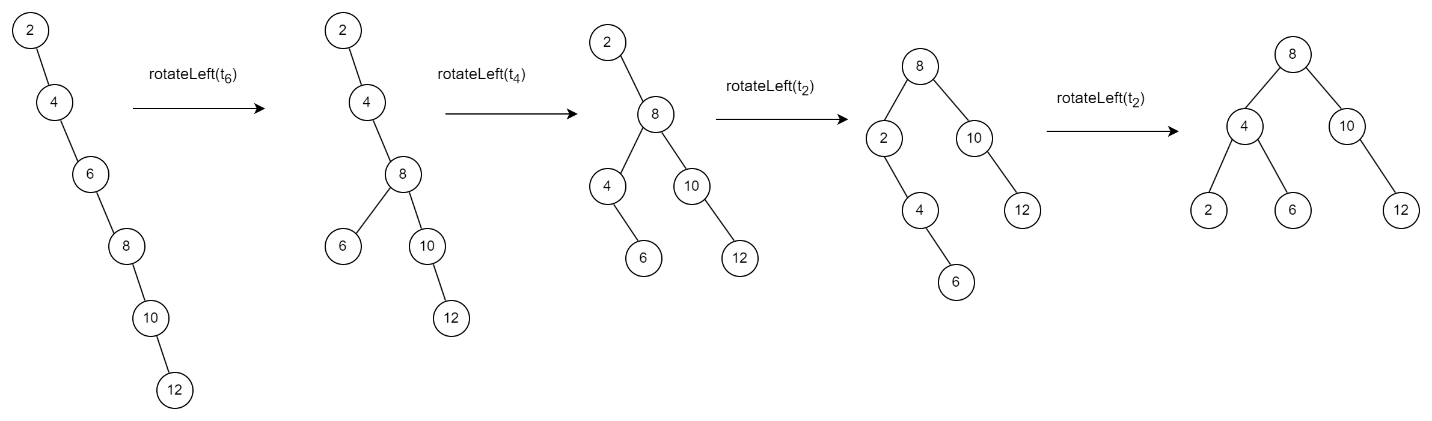
1. (Rebalancing)

Trace the execution of rebalance(t) on the following tree. Show the tree after each rotate operation.



Let be a tree rooted at node , and be the number of nodes in the tree . An operation to rebalance the tree begins by and successive recursive similar calls until the tree achieves a balanced state. In this case, a call to partition will involve a rotate operation that results into the following.

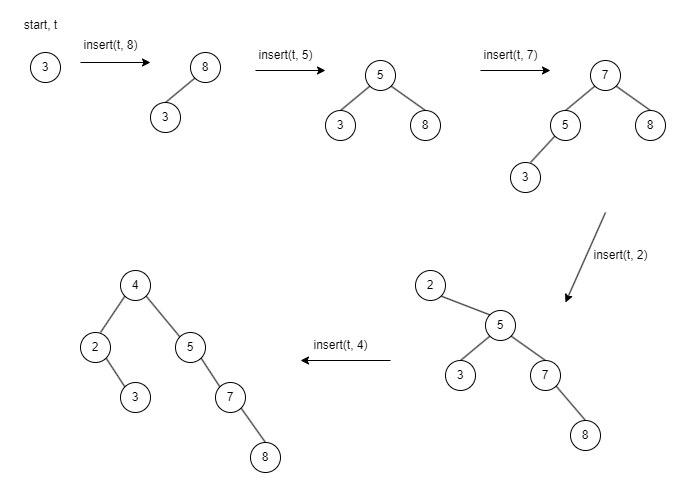


The first call to rotateLeft(t2) results into a sub-tree rooted at 2 with three nodes. This will require another rotation at root node 2 to achieve a balanced tree.

2. (Splay trees)

* 1. Show how a Splay tree would be constructed if the following values were inserted into an initially empty tree in the order given:

3 8 5 7 2 4



* 1. Let t be your answer to question a., and consider the following sequence of operations:

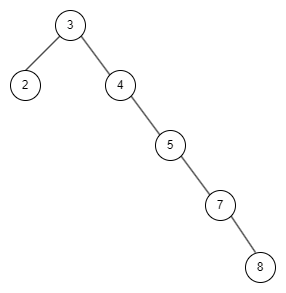
SearchSplay(t,3)

SearchSplay(t,5)

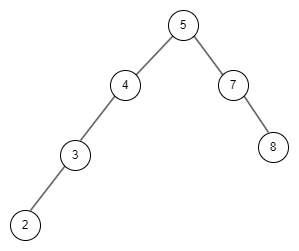
SearchSplay(t,6)

Show the tree after each operation.

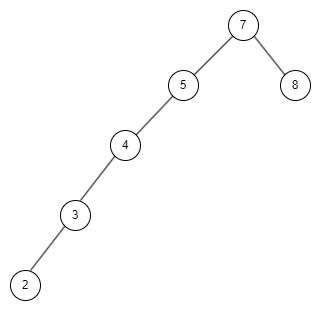
Below is the tree after the operation SearchSplay(t, 3)



Below is the tree after the operation SearchSplay(t, 5)



Below is the tree after operation SearchSplay(t, 6)



3. (AVL trees)

1. *Note: You should answer the following question without the help of the*treeLab*program from the lecture.*

Show how an AVL tree would be constructed if the following values were inserted into an initially empty tree in the order given:

58 26 12 37 43 40

1. xtend the BST ADT from the lecture ([BST.h](http://www.cse.unsw.edu.au/~cs9024/23T2/lecs/week8/progs/BST.h), [BST.c](http://www.cse.unsw.edu.au/~cs9024/23T2/lecs/week8/progs/BST.c)) by an implementation of the function