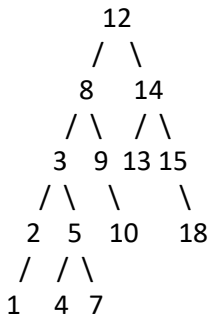
Part A:

1. Change the AVLtree class name to include your initials at the end.
   Write range method for a Binary Search Tree, which returns the difference between the minimum and maximum data, the range is 0 if the tree is empty.
2. Write rotateLeftChild, which takes an avlNode, and rotate with its Left Child, update the height and return the new node.
3. Write a program to print all given input names in the list, grouped by number of characters. The runtime must be O(nlogn) where n is the number of names.
   All names with the same number of characters should be printed on the same line, and names with fewer characters should be printed first.
   The runtime must be O(nlogn) where n is the number of names in the list.
   You may use following map to store the string to be printed on the same line.
   Map<Integer, List<String>> map = new ... // Choose the appropriate map class
   Note: Use a single String to store more than one name may slow down the runtime (but a StringBuilder is fine).
   Write the class name as A3PrintName follow by your initials.

Part B

1. What is the minimum number of node in an AVL tree with height of 3, height 5 and height 8? Write a program to compute.

Consider following AVL tree



2. How does the above tree look like after inserting 6 and 11.

3. Does the tree perform any rotation if we delete 3? If it does, which nodes and what rotation?

4. Does the tree perform any rotation if we delete the maximum value? If it does, which nodes and what rotation?

5. What is the worse-case runtime of deleting an element from an AVL tree? What about an unbalanced binary search tree?