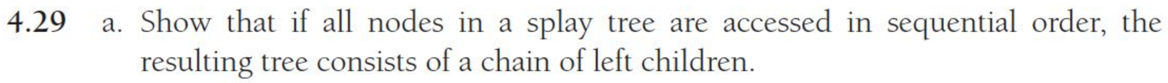
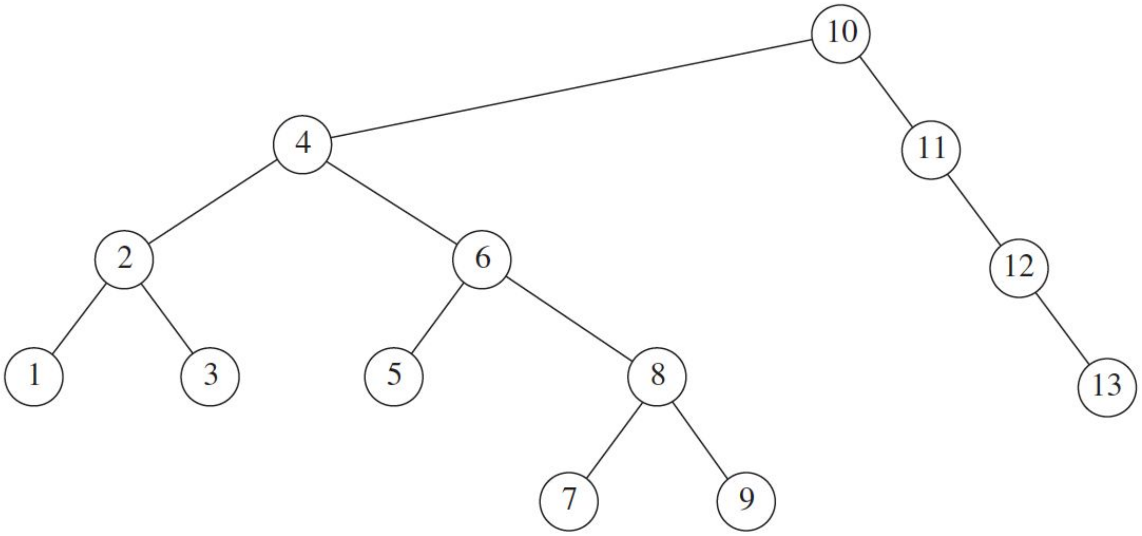
HW3

75 points
Textbook Questions: 4.29 (Written) and 4.23 (Java Program)
Due Thursday, October 11th, 11:59pm, submitted on Canvas

Written Question
4.29 (20 points)



4.29 a. Show that if all nodes in a splay tree are accessed in sequential order, the resulting tree consists of a chain of left children.



Draw 13 diagrams, 1 diagram after each node is accessed. You do not need to show all the rotation steps for each accessed node. Just draw the diagram after the node is accessed.

13 diagrams:

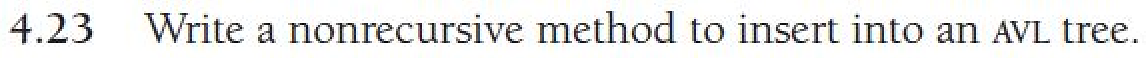
1. Show the tree after Node 1 is accessed (find(1)). The root will be be Node 1.
2. Show the tree after Node 2 is accessed (find(2)).
   ….

13. Show the tree after Node 13 is accessed (find(13)).

Draw the diagrams on a piece of paper, then include the .jpg files or word document in your zip file. Please make it clear and easy to view for the grader.

Java Program

4.23 55 points



4.23 Write a nonrecursive method to insert into an AVL tree.

I have provided you with the starter code AVLTree.java for you. You need to add a nonrecursive insert method to the code that will both insert the new node, and correctly balance the tree after an insert puts any node out of balance.

Write a TestAVLTree class which creates an AVLTree object, reads int values from the file "input.txt", and then inserts the values as nodes into the AVLTree using your new nonrecursive insert method. Print the tree to the console using a preorder traversal, to verify it's an AVL tree, and all the inserts and rebalances were done correctly.

You can use the recursive insert method, which is already included in AVLTree.java, while you are testing your new insert method. However, once your new insert method is complete, comment out the recursive insert method to make it easy for the grader.

Here is sample output from a preorder traversal of the AVL tree, after inserting all the values.

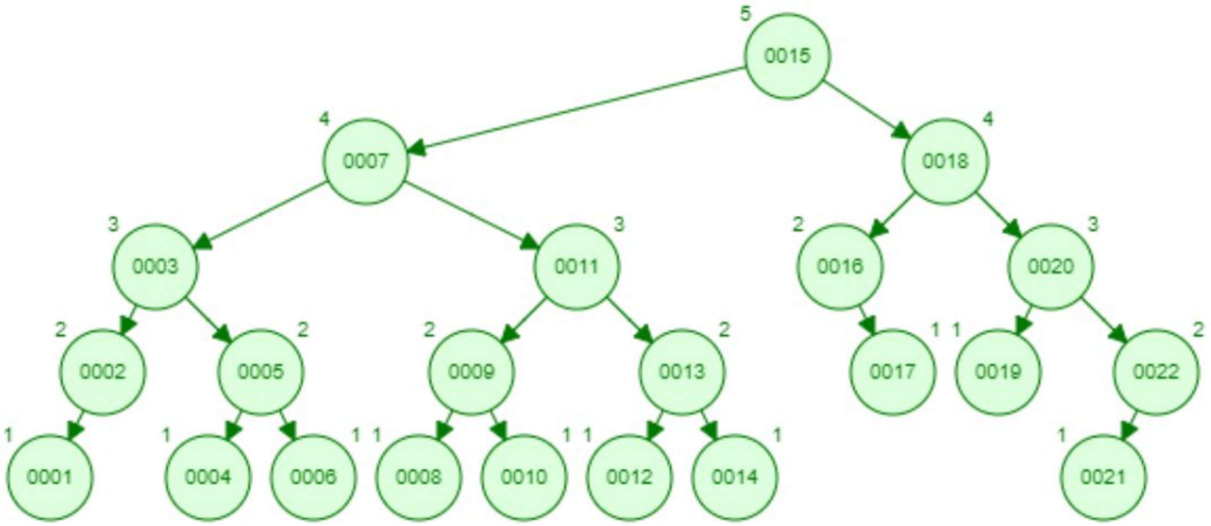
Input.txt

```
20 3 16 2 11 7 10 4 15 1 5 12 6 13 18 8 19 9 14 17 22 21
```

Program output:

```
15 7 3 2 1 5 4 6 11 9 8 10 13 12 14 18 16 17 20 19 22 21
```

Here's a visual of what the AVL Tree should look like a after the ints in input.txt are inserted:



The grader will use a different input.txt for verification.

Submit to Canvas:
Submit one zip file containing your src folder with your .java files and input.txt. The zip file will also contain your solution to the written question. Please make your zip file well organized for the grader.

Written Question Grading Criteria:
1.5 points for the first 12 diagrams. 2 points for the last diagram - 20 points total.

Program Grading Criteria	**% Points**
Program(s) fulfill all the requirements. All .java files are included and declare the necessary classes. Code is well organized, and easy to follow (especially for the grader). Coding stye is well utilized, including well named variables and methods. Comments are included, well written and descriptive.	100%
Program(s) fulfill almost all of the requirements. All .java files are included and declare the necessary classes. Code is fairly well organized, and somewhat easy to follow (especially for the grader). Some comments are included.	80%
Program(s) fulfill most of the requirements. All .java files are included and declare the necessary classes. Code is fairly well organized, and somewhat easy to follow (especially for the grader). Some or no comments are included.	60%
Program(s) fulfills some of the requirements, or does not run at all. Some .java files are included and declare some of the necessary classes. Some or no comments are included.	40%
Program(s) does not run at all. Some .java files are included and declare some of the necessary classes. Some or no comments are included.	20%
Either no attempt was made, or the attempt made shows no progress toward solving the problem.	0%