Lab Exercise: AVL Trees (paper exercise)

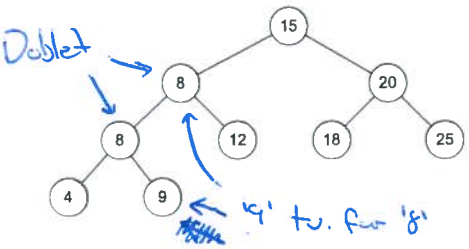
In this paper exercise you will gain some routine in the recognition of AVL trees and insertion and removal of elements in/from AVL trees

---

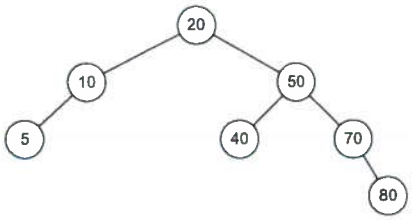
Exercise 1:
Determine for each of the following binary trees if they are BSTs and AVL trees:

Tree 1:

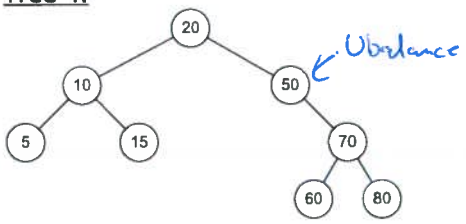
Tree 2:



Tree 3:



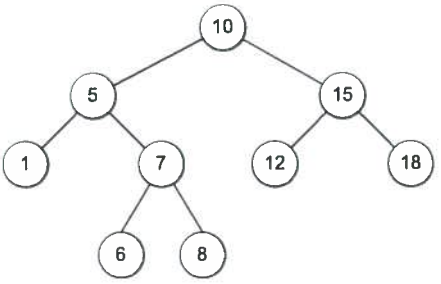
Tree 4:

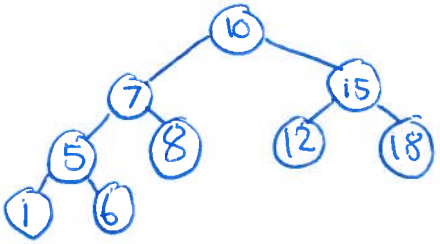


	Tree 1	Tree 2	Tree 3	Tree 4
BST	X		X	X
AVL	X		X	

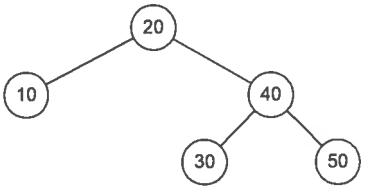
Exercise 2:
Insert 9 in the following AVL tree. Determine:

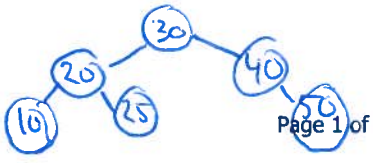
1. Where must 9 be inserted? Right child of '8'
2. Does this cause an imbalance anywhere? Yes, at '5'
3. What case of imbalance is caused? RR @ '5'
4. What do you do about the imbalance? LR @ '5'
5. What does the tree look like after the insertion?



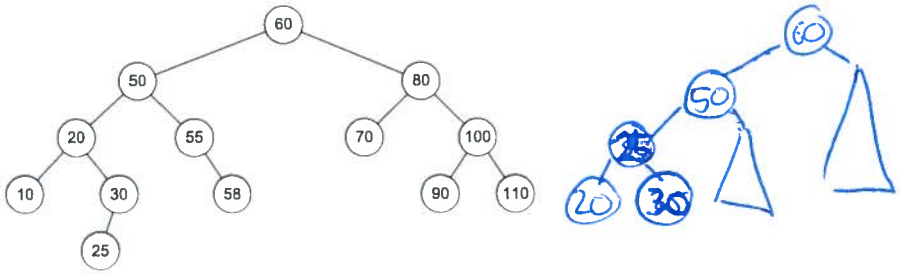


Exercise 3:
How does this AVL tree look after insertion of 25?
Use the same recipe as in exercise 2.
LR imbalance @ '20' ⇒ RL rotation @ '20'



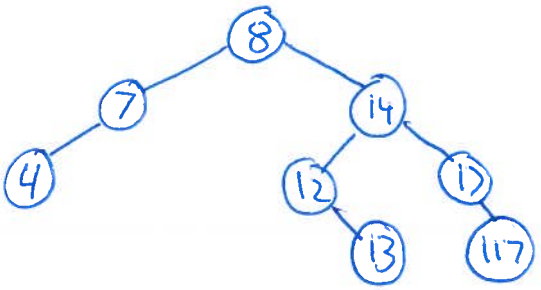


Exercise 4:
How does this AVL tree look after deletion of 10?



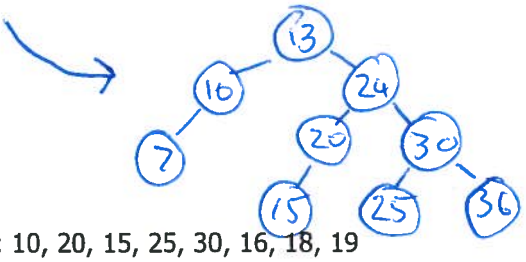
Exercise 5:
Perform the following operations in an initially empty AVL tree. Then show the final structure of the tree

1. insert(14)
2. insert(17)
3. insert(11)
4. insert(7)
5. insert(53)
6. insert(4)
7. insert(13)
8. insert(12)
9. insert(8)
10. remove(53)
11. remove(11)
12. insert(117)



Exercise 6:
Build an AVL tree by inserting the following values in the order stated: 15, 20, 24, 10, 13, 7, 30, 36, 25

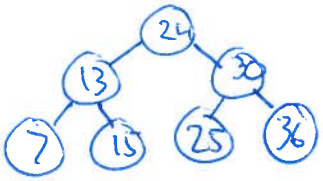
Exercise 7:
Remove 10 and 20 from the tree resulting from Exercise 6



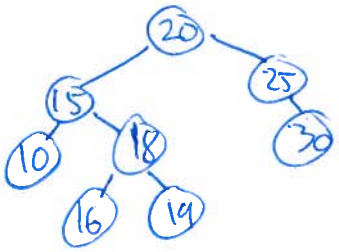
Exercise 8:
Build an AVL tree by inserting the following values in the order stated: 10, 20, 15, 25, 30, 16, 18, 19

Exercise 9:
Remove 30 from the tree resulting from Exercise 8

Ex. 7:



Ex. 8:



Ex. 9