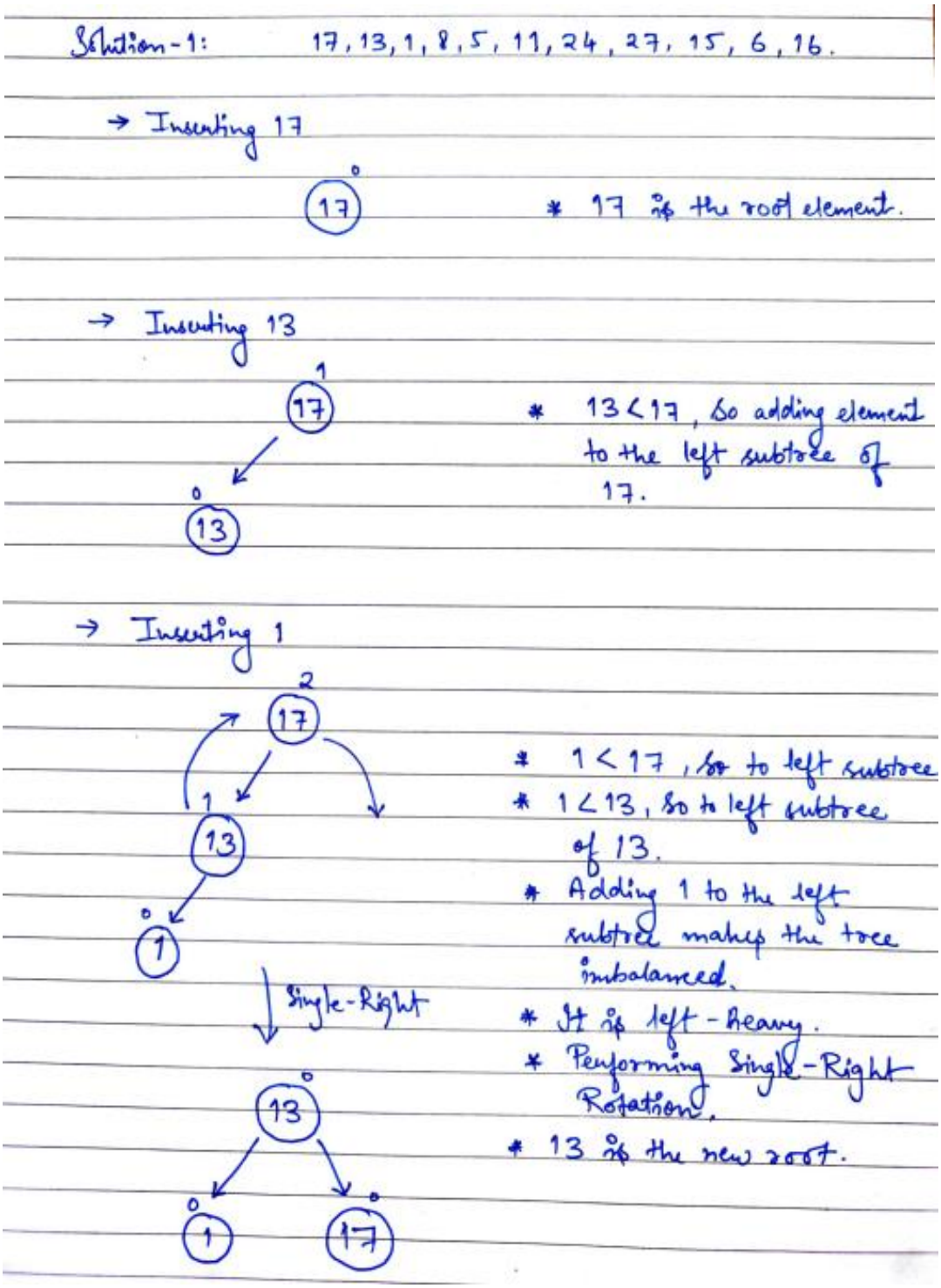
Task 7.1P

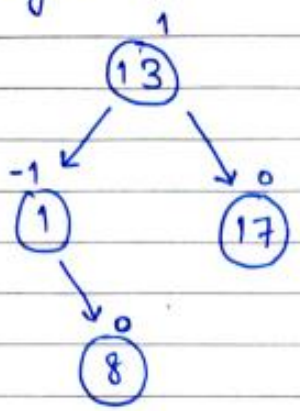
Name- Anneshu Nag
Student ID- 2210994760

Q1. Draw a series of figures demonstrating the insertion of the values. 17, 13, 1, 8, 5, 11, 24, 27, 15, 6, 16 into an initially empty AVL tree. Insert the values in the order they appear in the given sequence.

Ans:

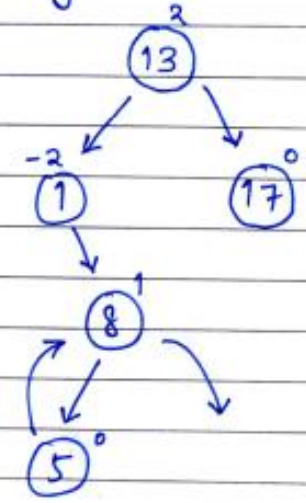


→ Inserting 8



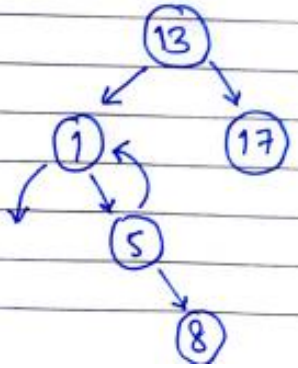
* 8 < 13, so to left subtree
* 8 > 1, so to right subtree
* 8 will be added to right subtree of 1.
* It is balanced.
* No rotation required.

→ Inserting 5.

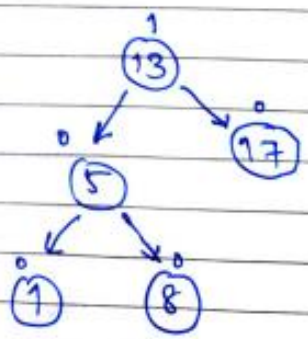


* 5 < 13, so to left subtree.
* 5 > 1, so to right subtree
* 5 < 8, so to left subtree
* 5 will be added to left subtree of 8.
* It is unbalanced & subtree of 1 is left heavy in the right subtree.
* Performing Right - Left Rotation.

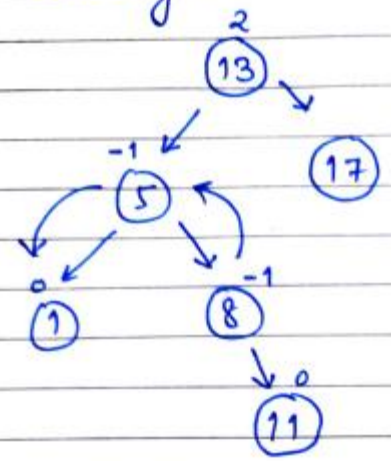
Right Rotation



Left Rotation

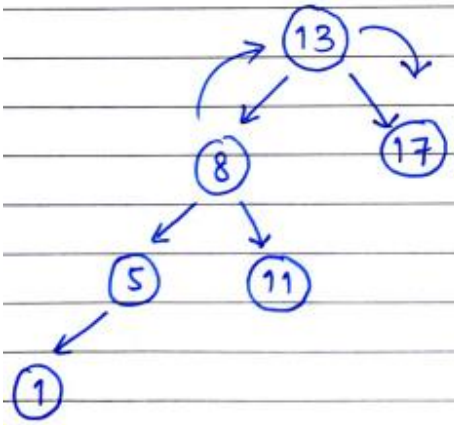


→ Inserting 11

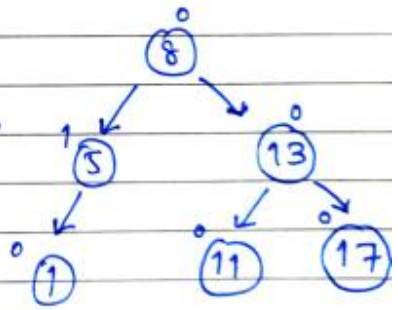


* 11 < 13, so to left subtree.
* 11 > 5, so to right subtree
* 11 > 8, so to right subtree.
* 11 will be added to right subtree of 8.
* It is unbalanced & subtree at the left of 13 is right heavy.
* Performing Left - Right Rotation.
* 8 is the new root.

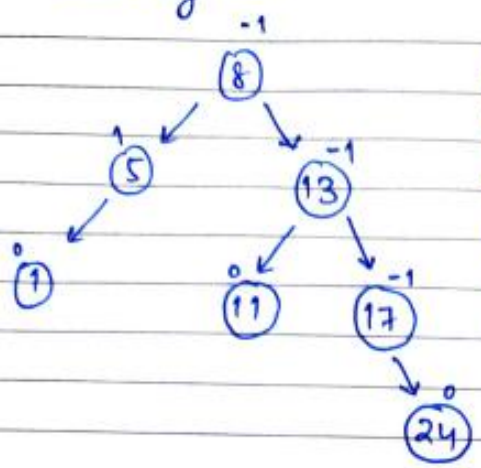
Left Rotation.



Right Rotation.

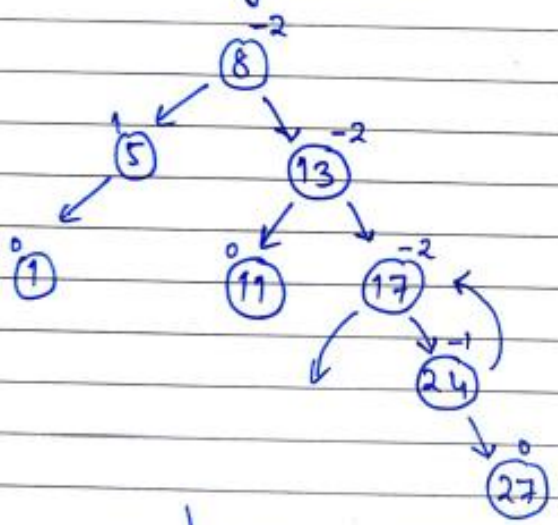


→ Inserting 24



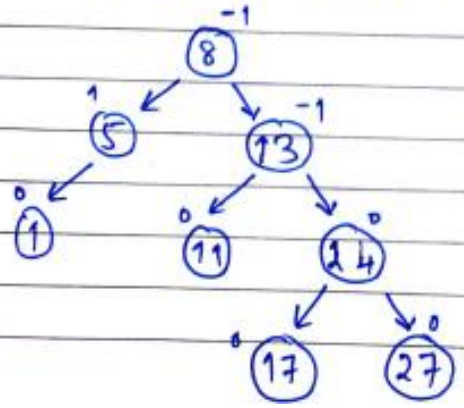
* 24 > 8, so to right-subtree.
* 24 > 13, so to right-subtree
* 24 > 17, so to right-subtree.
* It is balanced.
* No rotation required.
* 24 is added to right-subtree of 17.

→ Inserting 27

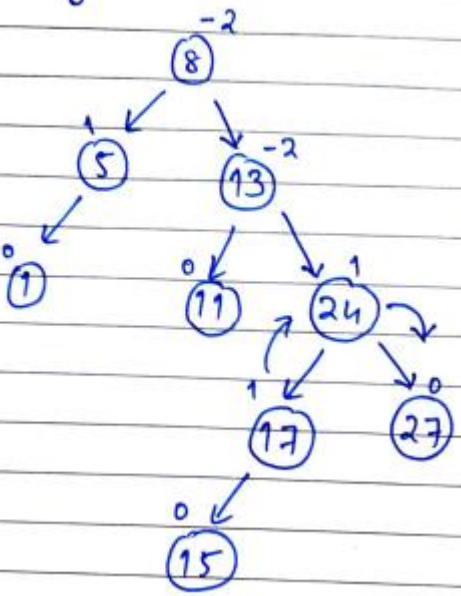


* 27 > 8, so to right-subtree.
* 27 > 13, so to right-subtree.
* 27 > 17, so to right-subtree.
* 27 > 24, so to right-subtree
* 27 is added to right subtree of 24.
* It is not balanced & the right subtree of 13 is right heavy.
* Performing Single-Left Rotation.

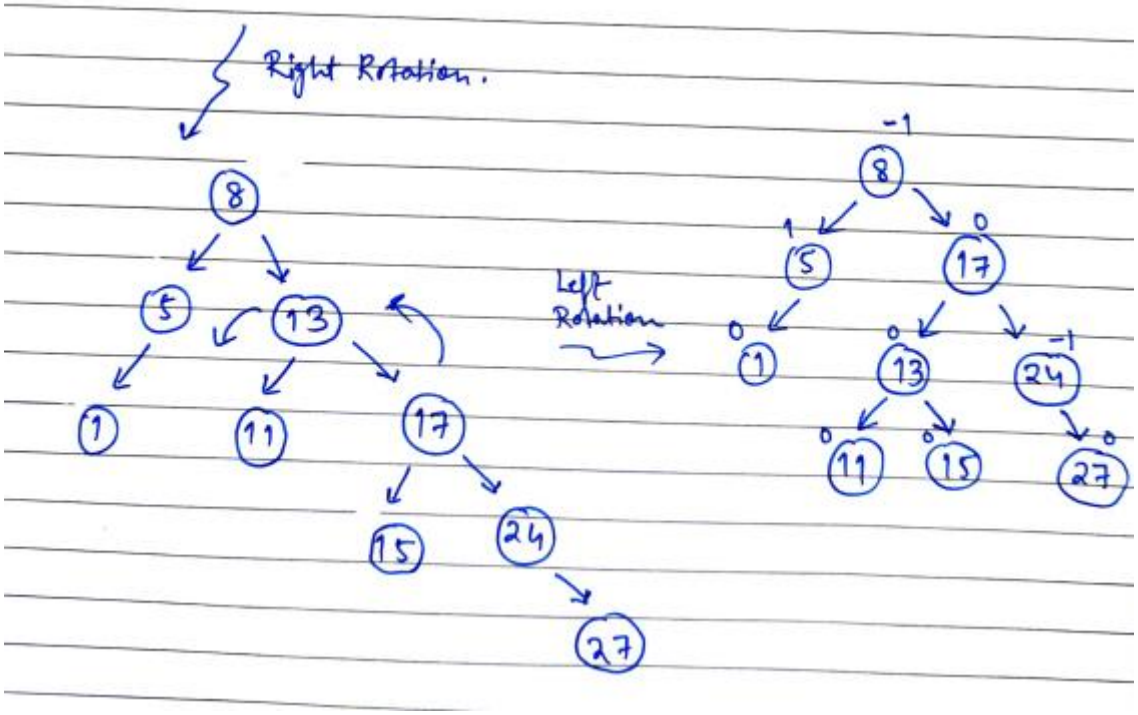
↓ Single Left Rotation.



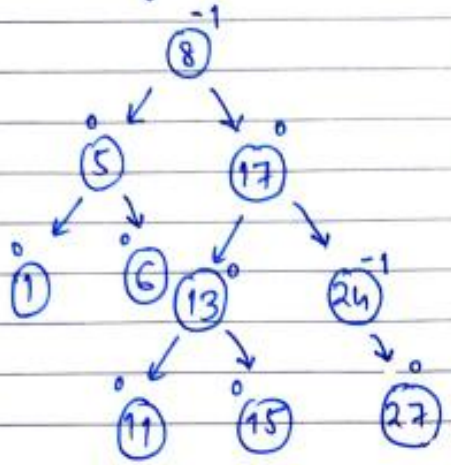
→ Inserting 15



* 15 > 8, so to right-subtree.
* 15 > 13, so to right-subtree
* 15 < 24, so to left-subtree
* 15 < 17, so to left-subtree.
* 15 will be added to left subtree of 17.
* It is not balanced & the right subtree of 13 is left heavy.
* Performing Right-Left Rotation.

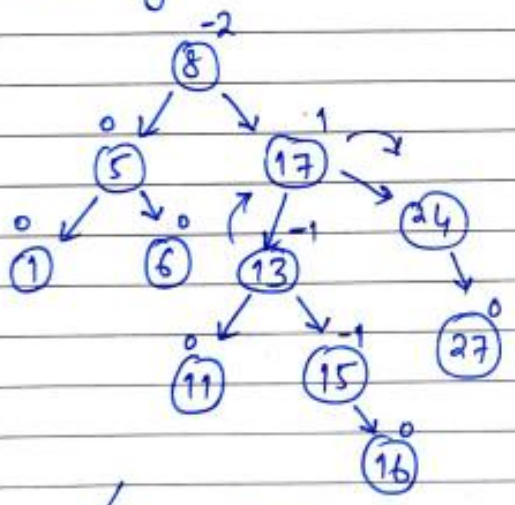


→ Inserting 6



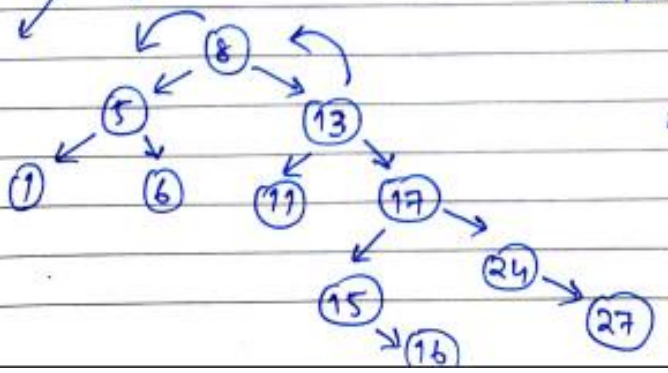
* $6 < 8$, so to the left-subtree
* $6 > 5$, so to the right-subtree
* 6 will be added to the right subtree of 5.
* It is balanced.

→ Inserting 16

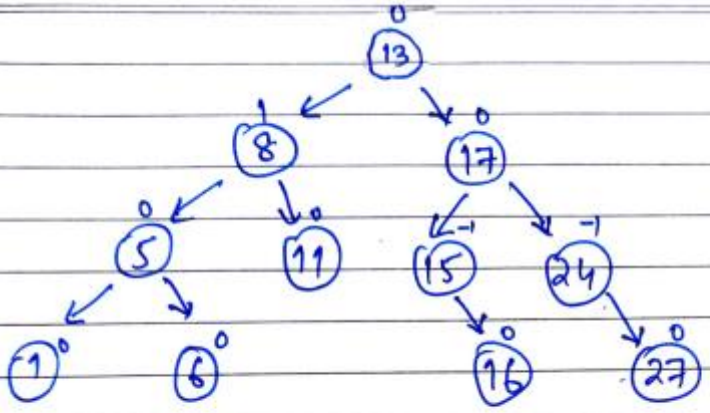


* 16 > 8, so to the right subtree
* 16 < 17, so to the left subtree
* 16 > 13, so to the right subtree
* 16 > 15, so to the right subtree
* 16 will be added to the right subtree of 15.
* It is not balanced & the right subtree of 8 is left heavy.
* Performing Right-Left Rotation.

Right Rotation.



Left Rotation.



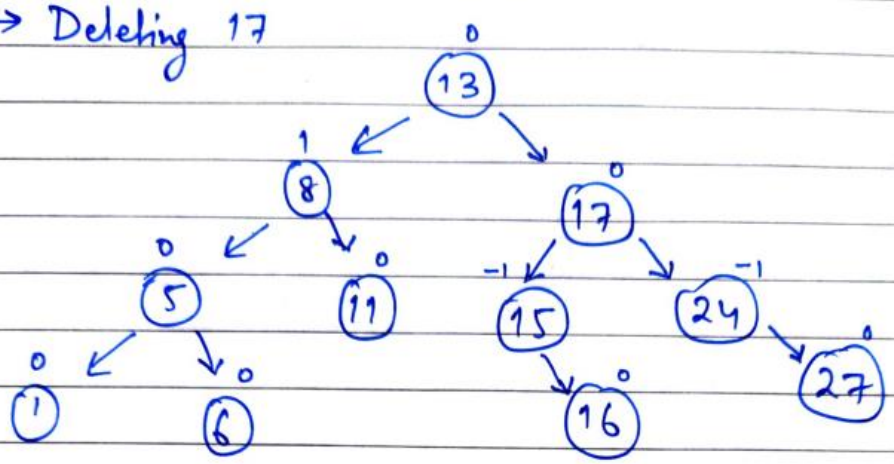
Hence the AVL tree is complete.

Q2. Now, draw a series of figures showing the deletion of the values: 17, 15, 11, 24, 27, 13, 16, 5, 6, 1, 8 from the AVL tree built in the previous part of the task. Delete the values in the order they appear in the given sequence. Draw the AVL tree after each deletion and rotation (if any), and complement each of the figures with all the aforementioned details.

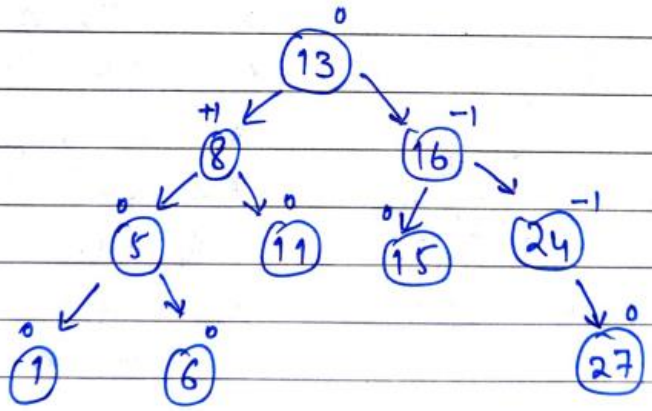
Ans:

Solution-2: 17, 15, 11, 24, 27, 13, 16, 5, 6, 1, 8

→ Deleting 17

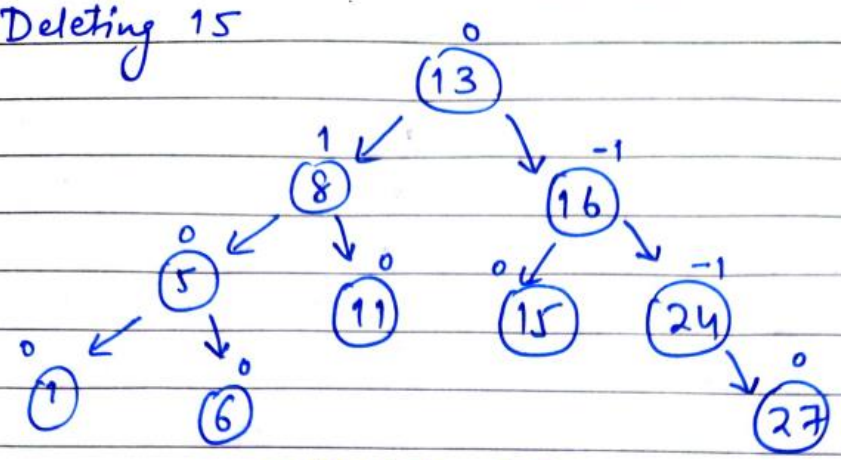


* Swap (17 & 16)
* Delete 17

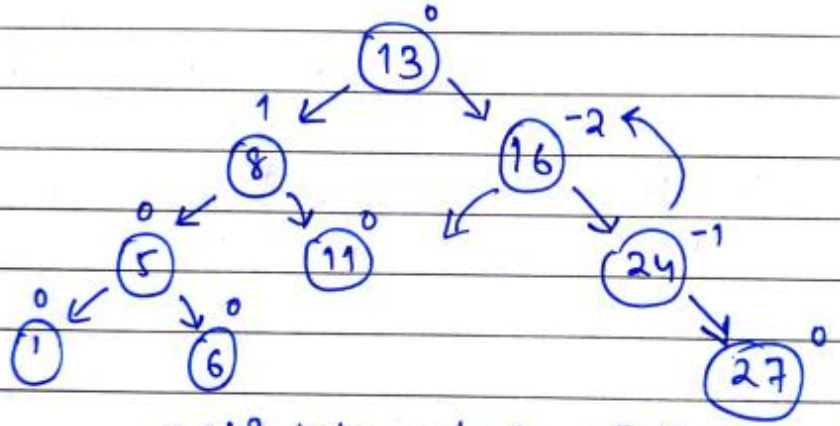


* It is balanced. So no rotations required.

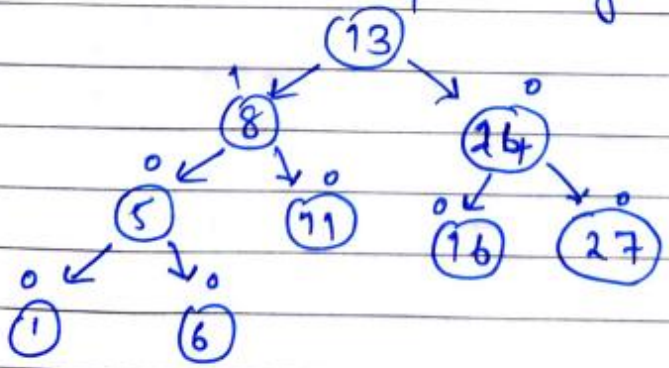
→ Deleting 15



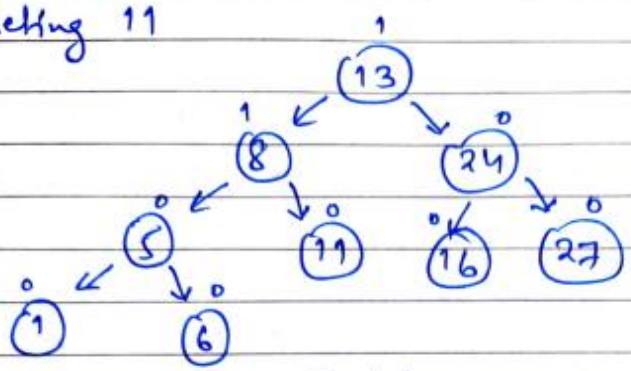
* Delete 15



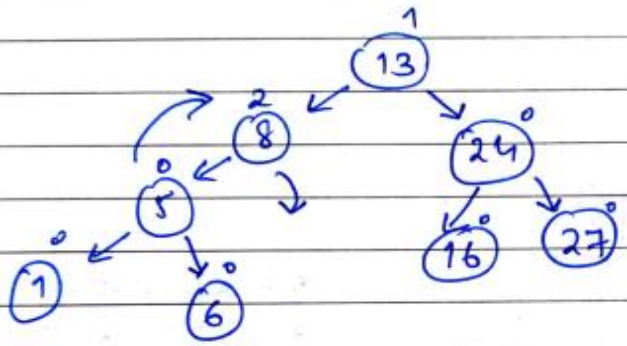
* Not balanced so rotation required.
* Rotating 16 to left. (Left Rotation) as it is right heavy.



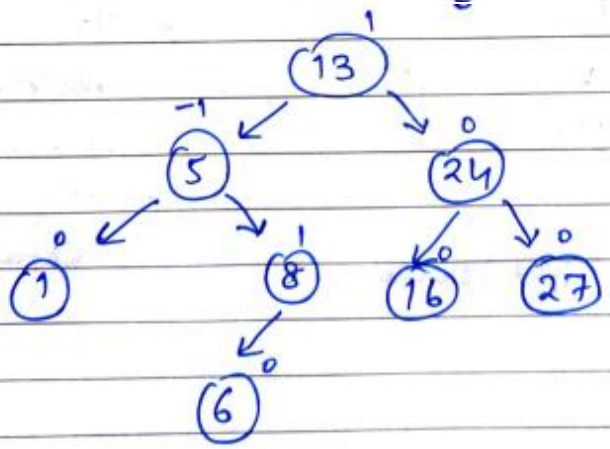
→ Deleting 11



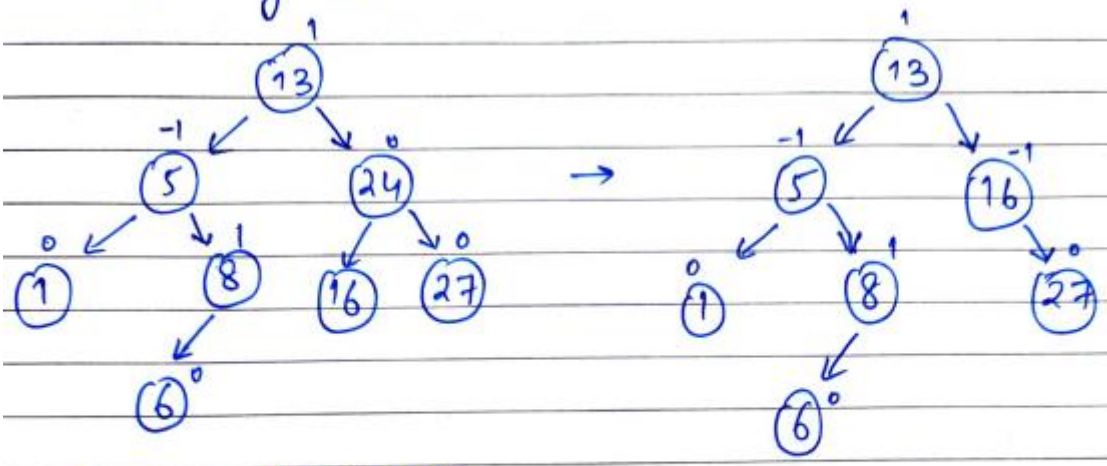
* Delete 11



* Not balanced so rotation required
* Rotating 8 to right (Right Rotation) as it is left heavy.

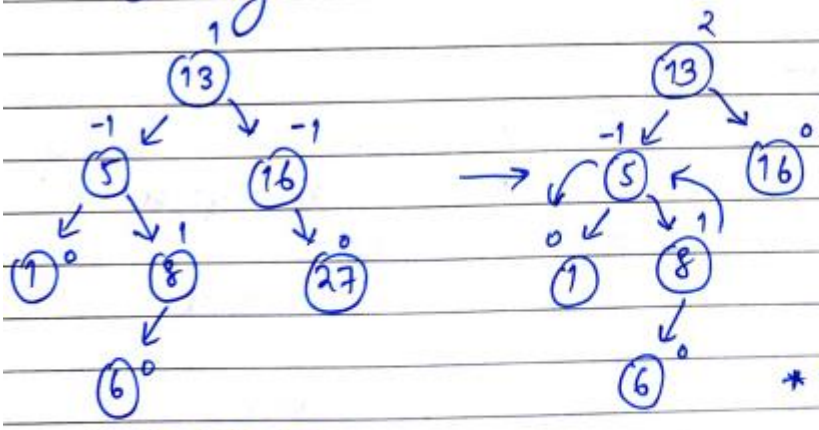


→ Deleting 24.



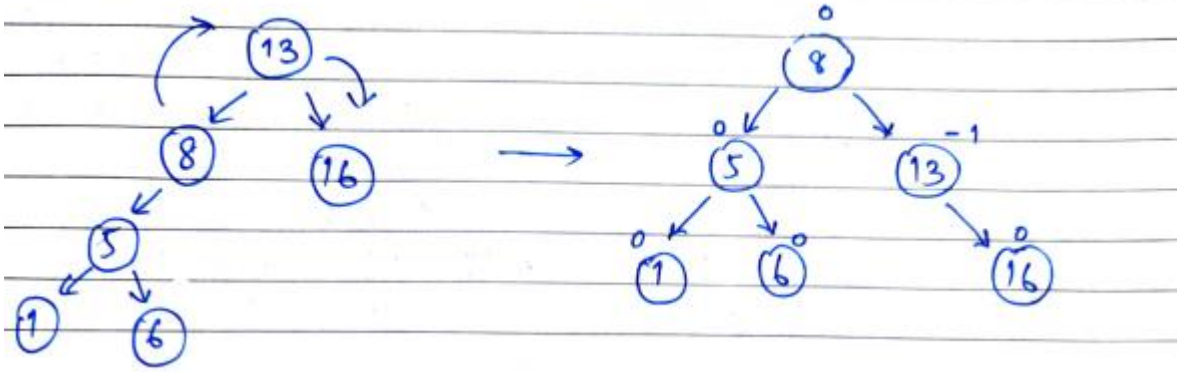
* Swap (24 & 16)
* Delete 24
* It is balanced so no rotation required.

→ Deleting 27

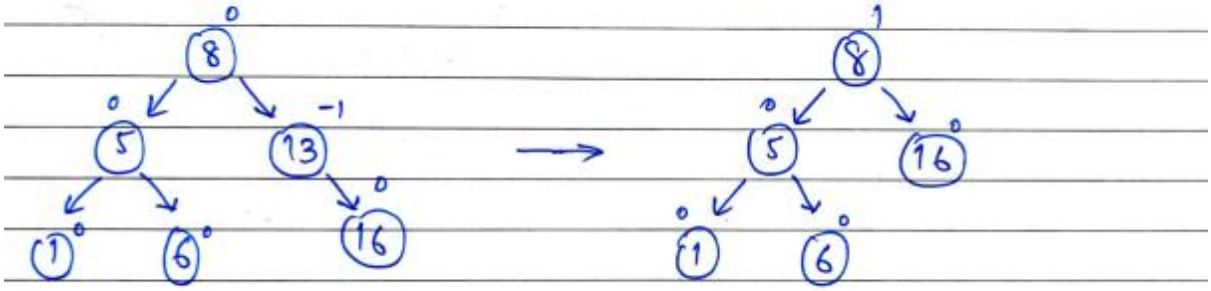


* Delete 27

* Not balanced so rotation required.
* Left-Right rotation required as left subtree of 13 is right heavy.



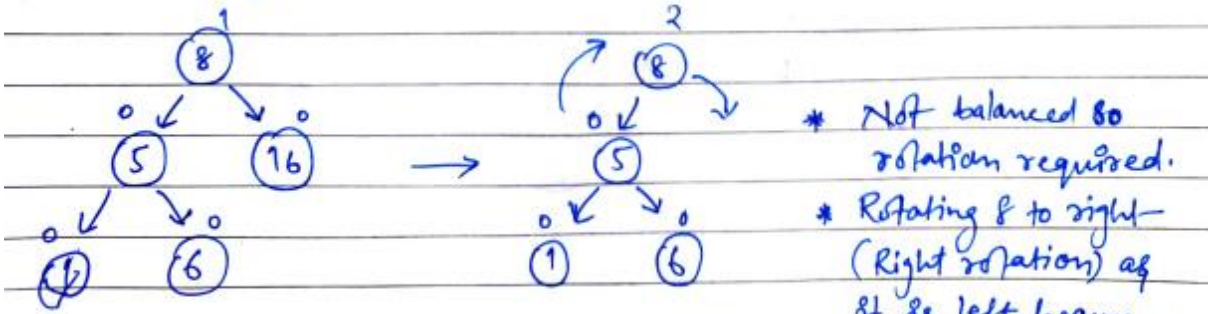
→ Deleting 13.



* Swape (13 & 16)
* Delete 13

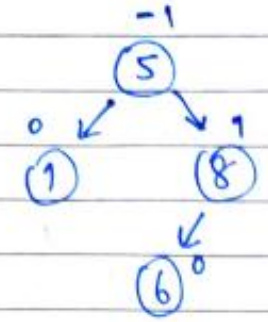
* It is balanced so no rotation required.

→ Deleting 16.

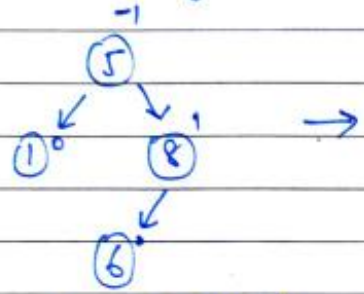


* Delete 16

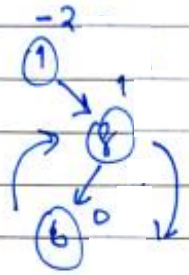
* Not balanced so rotation required.
* Rotating 8 to right (Right rotation) as it is left heavy.



→ Deleting 5

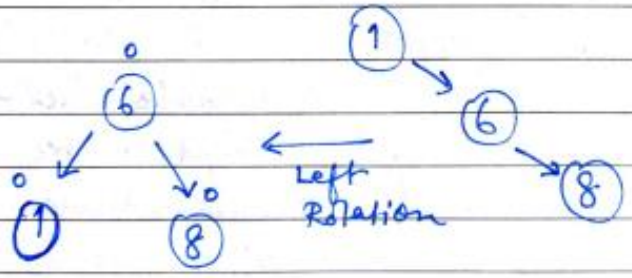


* Swap (5 & 1)
* Delete 5

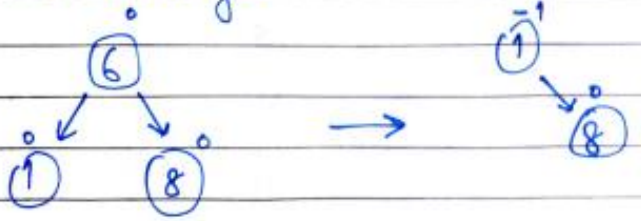


Right Rotation

* Not balanced so rotation required.
* Right - Left Rotation required as right subtree of 1 is left heavy.



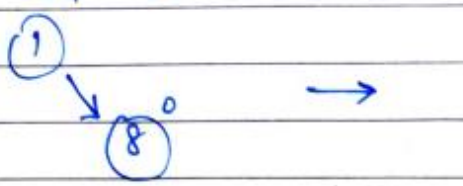
→ Deleting 6



* Swap (6 & 1)
* Delete 6

* Balanced
* No Rotation Required.

→ Deleting 1



* 8 is the root element.

* Swap (1 & 8)
* Delete 1

→ Deleting 8

* Delete 8

Hence, All the elements are removed from the AVL tree.

3. What order should we insert the elements {1, 2, … ,7} into an empty AVL tree so that we do not have to perform any rotations on it?

Ans:

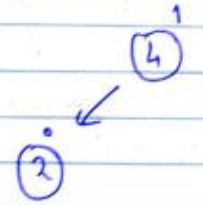
Solution - 3:

Inserting {1, 2, ... , 7}
Order = ?
Requirements : No performing any rotation.

→ Inserting 4

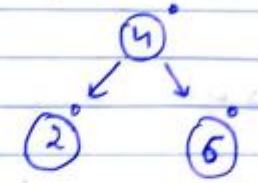
* 4 will be root element

→ Inserting 2



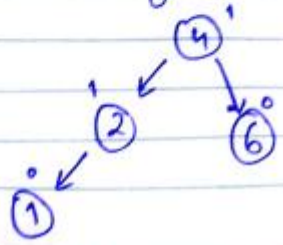
* 2 < 4, so it will be added to left subtree.
* It is balanced so no rotations performed.

→ Inserting 56



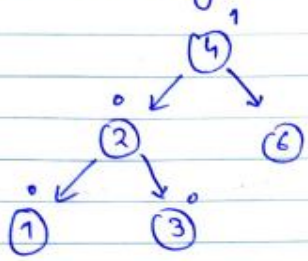
* 6 > 4, so it will be added to right subtree
* It is balanced so no rotations performed.

→ Inserting 1



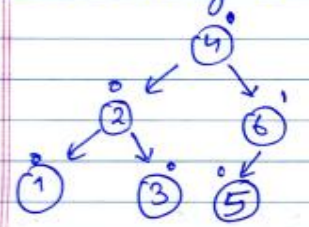
* 1 < 4, so to the left subtree
* 1 < 2, so to the left subtree
* 1 will be added to the left of 2
* It is balanced so no rotations performed.

→ Inserting 3



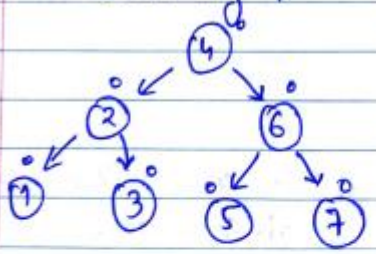
* 3 < 4, so to left subtree.
* 3 > 2, so to right subtree.
* 3 will be added to right subtree of 2.
* It is balanced so no rotation required.

→ Inserting 5



* 5 > 4, so to right subtree
* 5 < 6, so to left subtree.
* 5 will be added to left subtree of 6.
* It is balanced so no rotation required.

→ Inserting 7



* 7 > 4, so to right subtree
* 7 > 6, so to right subtree
* 7 will be added to right subtree of 6
* It is balanced so no rotation required.

Hence the AVL is completly formed without any rotations on it.

∴ Order = 4, 2, 6, 1, 3, 5, 7

* There can be other orders. *