

CS 202 Homework 03 Report

AVL Tree – Heap Implementation

Ali Eren Günaltılı

CS 202- Section 01

21801897

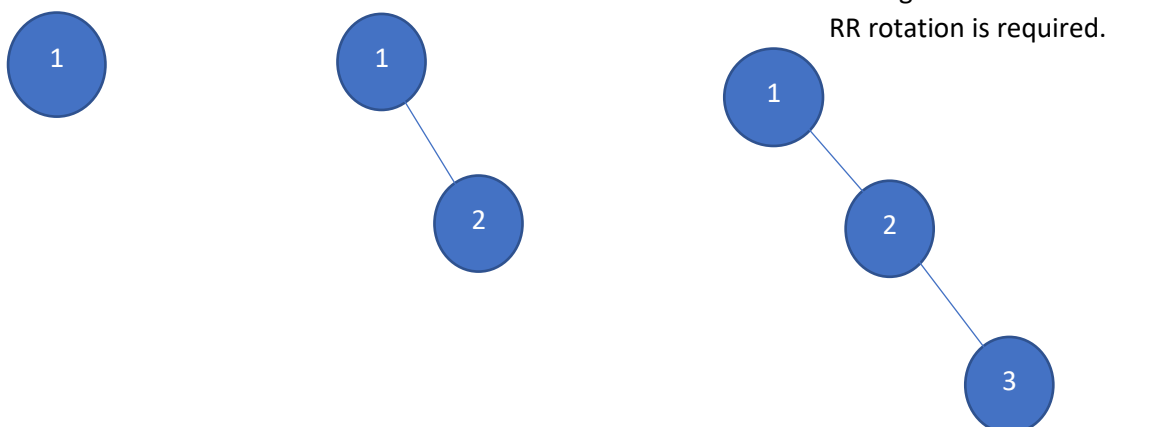
Output Table

-----TREE-----			
Array Size	Random	Ascending	Descending
1000	442	990	990
2000	1418	2979	2979
3000	2805	5967	5967
4000	4624	9955	9955
5000	6965	14942	14942
6000	9754	20929	20929
7000	12991	27916	27916
8000	16741	35903	35903
9000	20872	44889	44889
10000	25456	54875	54875

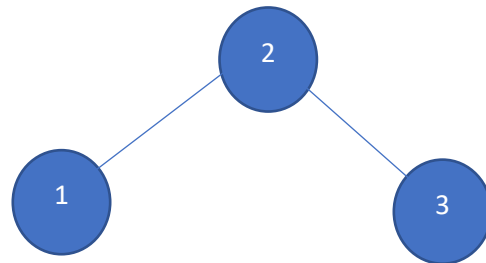
- Do your findings related to average number of rotations in the AVL tree agree with theoretical results?
- Do different patterns of insertion affect the number of rotations in the AVL tree? If so, explain how. If not, explain why not.

The number of rotations increases with respect to the array size just like expected. More insertion causes more rotation in the AVL tree basically. The real difference occurs between randomly created arrays and the other two arrays. Randomly created arrays have less rotation than the other two all the time. This is also another expected result. Since the worst case of AVL insertion occurs in ascendingly sorted or descendingly sorted trees. Insertion of randomly created values causes less rotation because sometimes they cannot change the balance of a tree as the value is randomly chosen. Insertion of ascending order and descending order values change the height of the AVL tree. Therefore, we are facing an unbalanced tree after a few steps. Descending order and ascending order insertion have the same number of rotations because problems occur in the same steps.

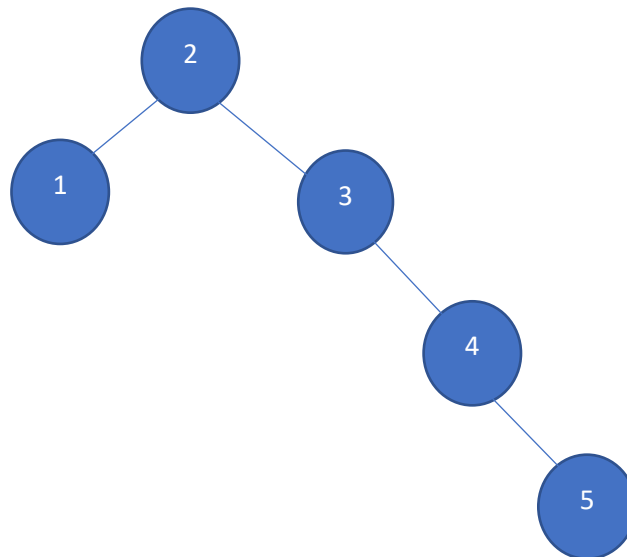
Ascending Order (Insertion of 1,2,3,4,5)



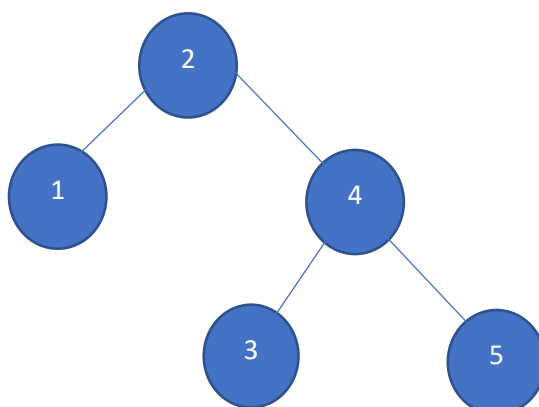
After RR rotation has done for tree.



Inserting 4 is not problematic but inserting 5 also cause RR rotation. For 3-4-5.



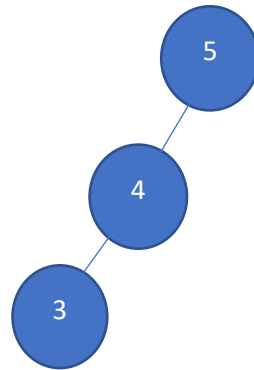
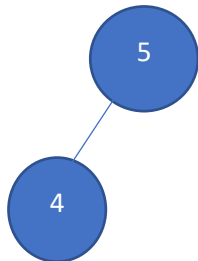
After RR rotation is done.



Adding 6 is also problematic cause RR rotation on 2-4-5.

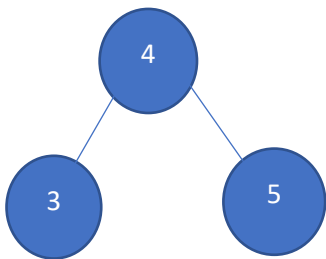
To sumup, in ascending order insertion we are facing with RR rotation constantly. But if numbers are given randomly, we may have a chance to insert without making the tree unbalance.

Descending Order(Insertion of 5,4,3,2,1)

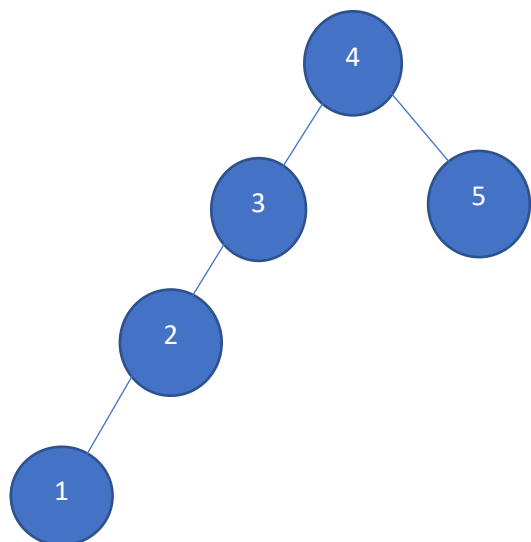


Insertion of 3 cause problem.

Insertion of 3 makes the tree unbalanced and requires LL rotation.



Insertion of 2 won't cause problem but 1 causes a problem.



1 cause problem and LL rotation is needed for 3-2-1.

In case of inserting 1,2,3,4,5 and 5,4,3,2,1 we need same number of rotation is same steps. The only difference is in ascending order we use RR rotation but in descending order we use LL rotation.

Another important point is that, randomly created insertion can change as values can change the number of rotations. But ascending order insertion and descending order insertion give the same result every time. Because, only thing that matter is their place in the AVL tree. That means, it doesn't matter whether we insert 1-2-3-4-5 or 1-6-8-11-14. AVL tree treats equally because both of them are in ascending order.

Another output

```
-----TREE-----
```

Array Size	Random	Ascending	Descending
1000	466	990	990
2000	1404	2979	2979
3000	2934	5967	5967
4000	5117	9955	9955
5000	8033	14942	14942
6000	11624	20929	20929
7000	16008	27916	27916
8000	21257	35903	35903
9000	27304	44889	44889
10000	34181	54875	54875