Bilkent University

CS 202

Fundamental Structures of Computer Science II

FALL 2021

3rd Homework

Heaps And AVL Trees

Name: Ferhat

Surname: Korkmaz

Section: 01

ID: 21901940

First, I want to show my output and a graph regarding the number of rotations occurring in an AVL Tree while inserting integer items into it.

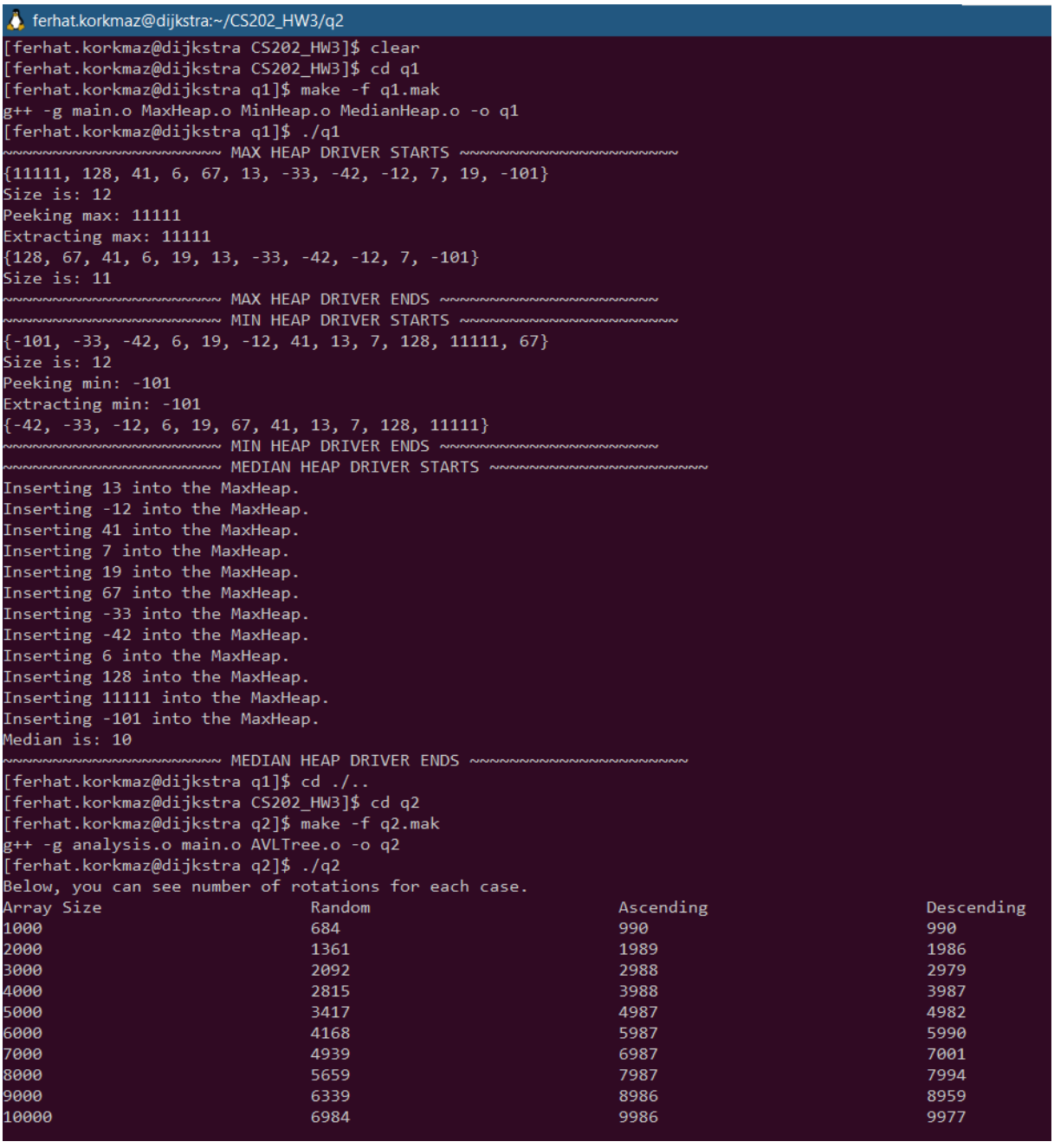


Figure 1: BCC Dijkstra Output

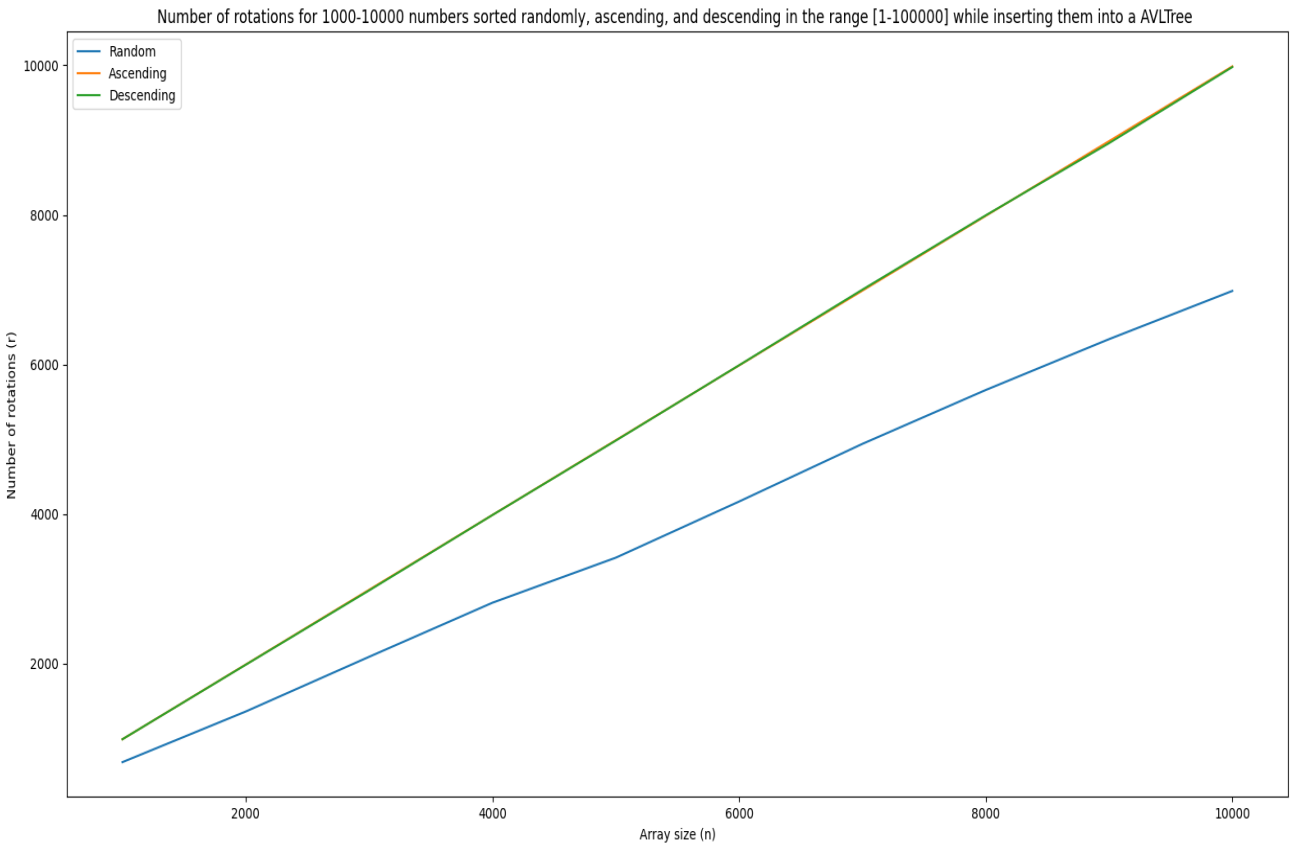


Figure 2: A graph demonstrating the data in my output.

Note that ascending and descending order arrays are so much similar such that they are on top of each other.

Different patterns affect the number of rotations as you can see from the output and the graph. Here the idea is very simple. When we have ascending list, the next number that is going to be inserted after the previous one will almost always be to the right of the previous one. This requires a rotation practically always. The case is very similar when the list is in descending order. Instead of keep inserting to the right of the parent node, now, we will insert to the left of it, which requires a rotation almost always. However, when the list is shuffled, we can expect that it is distributed more balanced, which will result in less number of rotations.

Here, I want to solve an example in which inserting the numbers [1-8] randomly and in ascending order into an AVL Tree to show the rotation number difference for these cases.

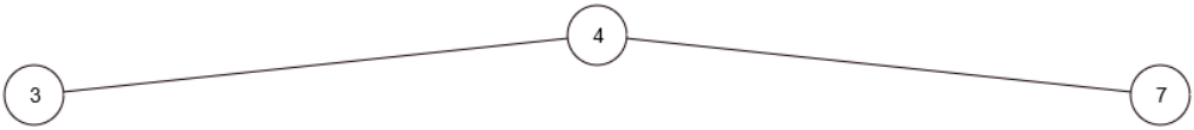
Array1 = {4, 3, 7, 1, 5, 2, 8, 6}

Array2 = {1, 2, 3, 4, 5, 6, 7, 8}

Let me consider the random array (Array 1) first step by step.

Inserting 4.

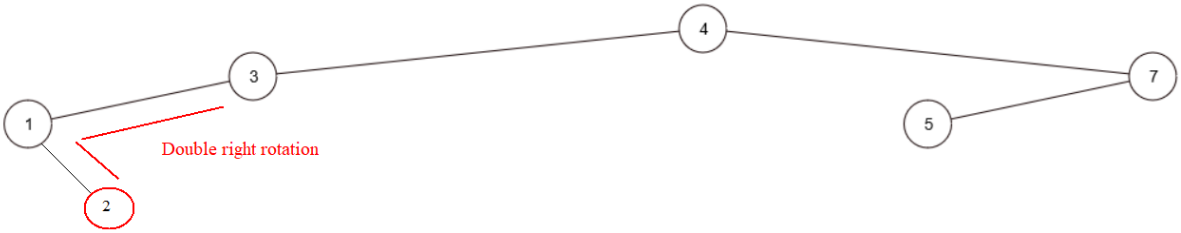
Inserting 3.



Inserting 7.

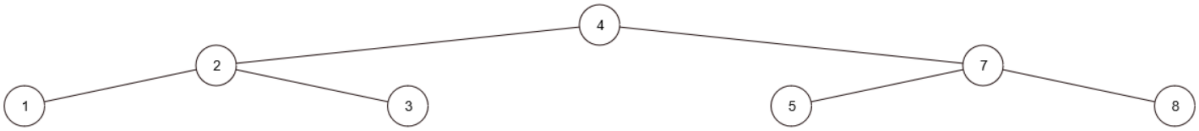
Inserting 1.

Inserting 5.

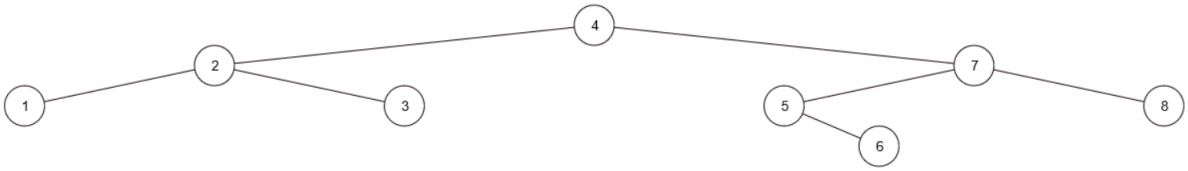


Inserting 2. Now we need to rotate twice to the right due to the node containg 3's height difference being more than 1.

After rotation.



Inserting 8.



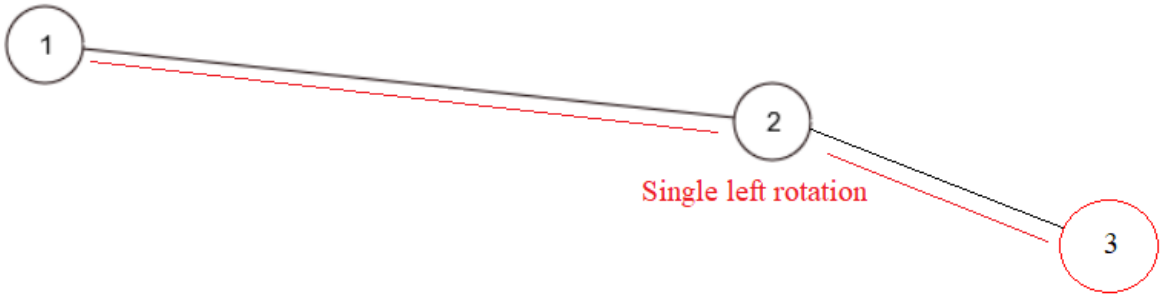
Inserting 6.

Total Rotation: 2.

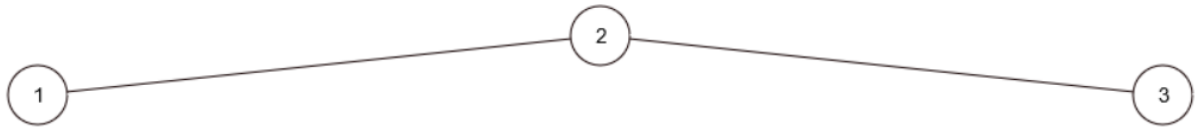
Now, let me consider the ascending array (Array 2) step by step.

Inserting 1.

Inserting 2.

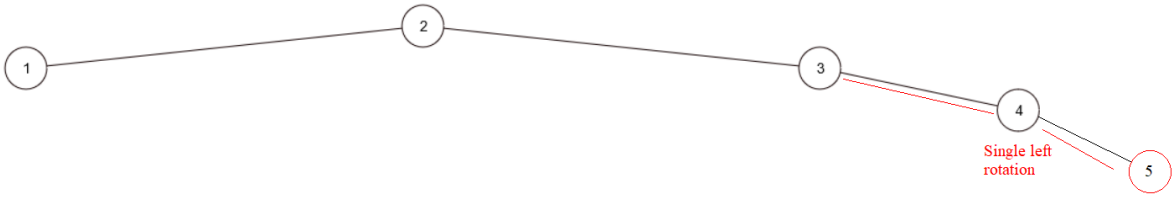


Inserting 3 requieres single left rotation. Due the node containg 1's height difference being more than 1.



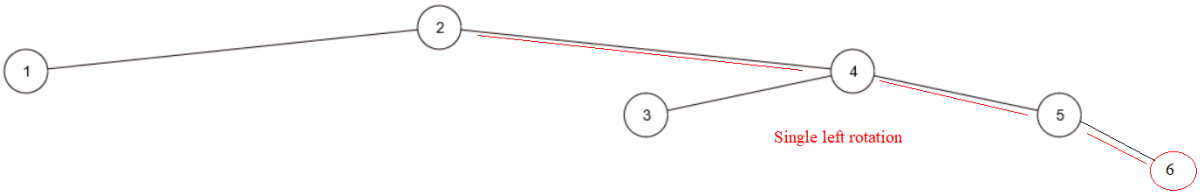
After balancing the AVL Tree.

Inserting 4.

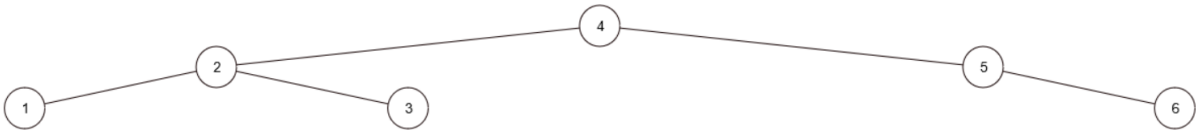


Inserting 5 requieres single left rotation due to the node containing 3's height difference being more than 1.

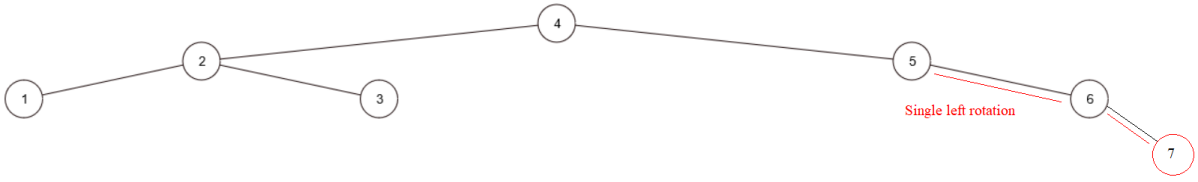
After balancing the AVL Tree.



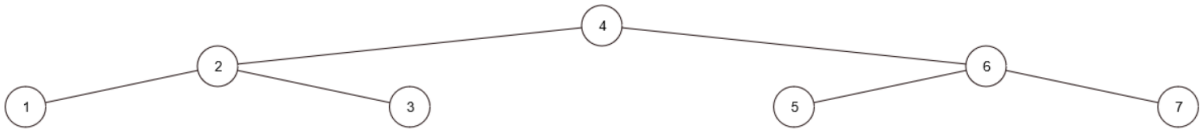
Inserting 6 requieres single left rotation due to the node containing 2's height difference being more than 1.



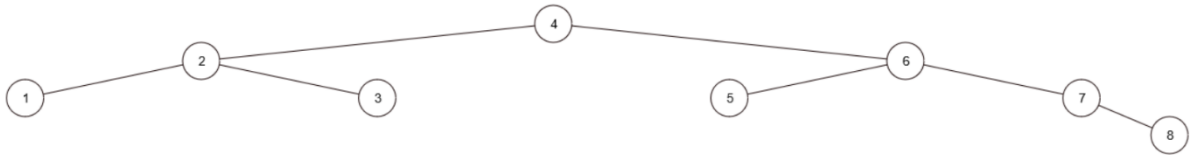
After balancing the AVL Tree.



Inserting 7 requieres single left rotation due to the node containg 5's height difference being more than 1.



After balancing the AVL Tree.



Inserting 8.

Total rotation: 4.

In lights of the experiemnt and how inserting logic works for AVL Trees, it was totally expected that my AVL Tree implementation’s rotation count would be noticeably smaller for the random created numbers by guessing the theoretical results for rotation.