# DSA456V1A

# Lab 7

# Student: Chungon Tse

# ID: 154928188

# Date: 19 Apr 2023

# BST and AVL

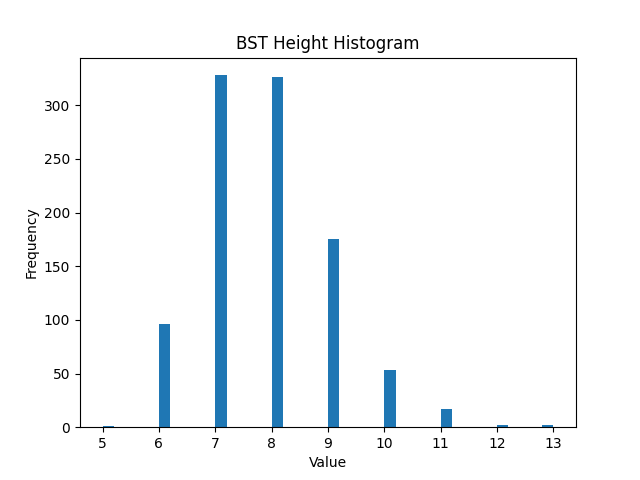
My rendition of code for a binary search tree generates figure 1 and figure 2 as resultant histograms about exactly 1000 random sequences using permutations of numbers 1-20 for the BST. I ran BST multiple times and got consistent results.

Figure 1

In figure 1, with height(), the two most common BST heights were 7 and 8, each happening in over 300 sequences. The extreme heights of 11-13 occurred in around 20 sequences.

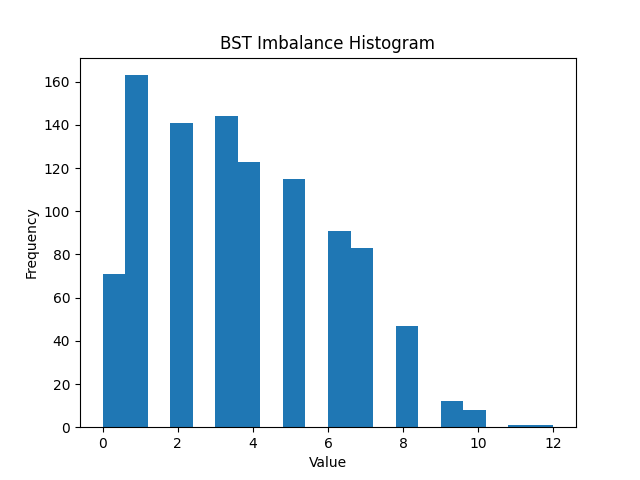


Figure 2

In figure 2, the most common BST imbalances were 1, 2, and 3, and each accounted for around 140-160 sequences. The most extreme imbalances of 10-12 happened in around 10 sequences.

Evidently, there are considerable variation in BST heights and a high frequency of imbalances above 1 in more than 200 sequences. I postulate that an AVL algorithm may be able to address this situation.

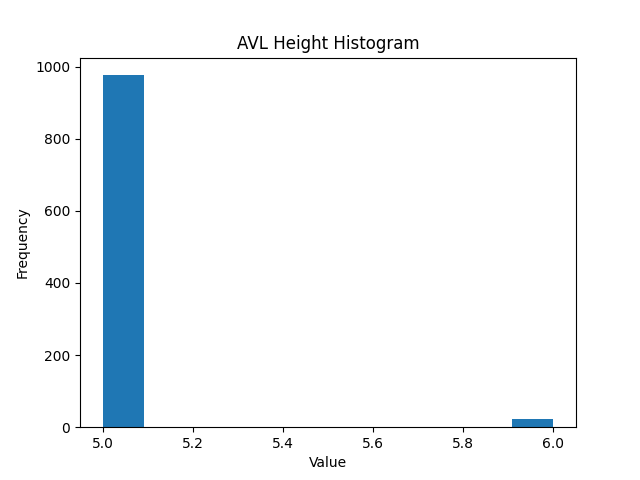


Figure 3

After applying an AVL algorithm, the vast majority of AVL trees are of height 5 and a few are of 6, as in figure 3.

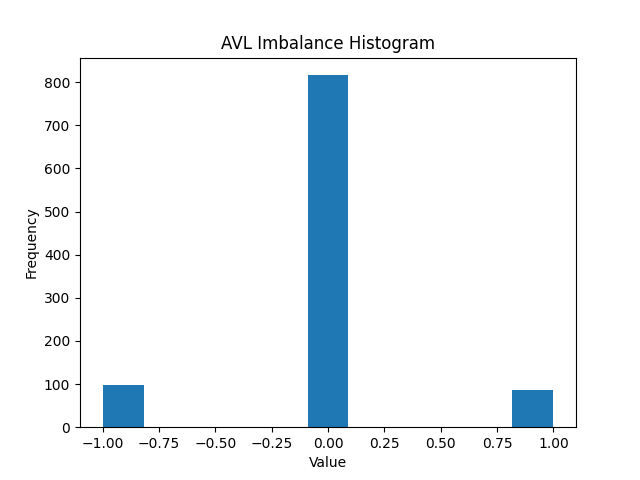


Figure 4

In figure 4, over 800 AVL trees were perfectly balanced. The minority were at -1 or +1 which was acceptable.

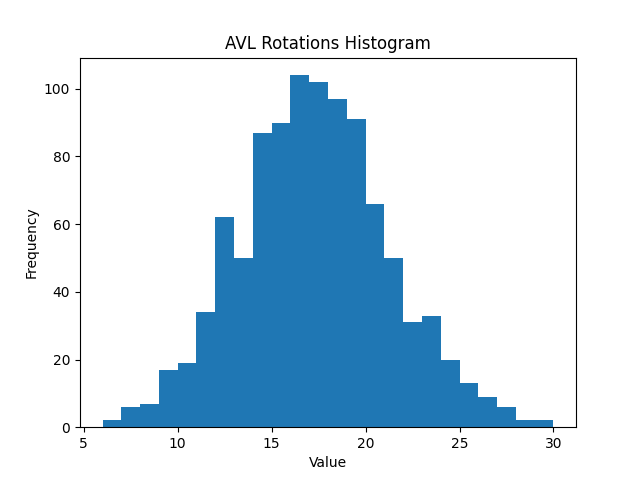


Figure 5

Figure 5 shows that the most sequences took 15 to 20 rotations. A small minority take just over 5 rotations or as many as 30 rotations.

Since the sequences are randomized, the above is just a snapshot of a trial run. However, exactly the algorithm is likely to generate similar results and consistently points out how AVL can greatly optimize BST height and reduce imbalances.