Lab 5 lab report

Name: Peter Haynes
Student id: prh7yc
Date: 10/17/19
Section: Tuesdays 12:30

**Comparing testfile1, testfile2, and testfile3:**

An AVL tree produces a balanced binary search tree. Because the tree is balanced, this creates more uniform run times and lowers the asymptotic run time. While in some cases binary search trees may find their elements faster, the worst case scenario has n run time. Imagine that each value added is always greater than the previous value, in this case, the BST would essentially operate as a linked list to find the desired answer. However, an AVL tree accounts for this possibility by constantly rebalancing. A great depiction of this is in the first testfile. Since the first element, “We”, is late in the alphabet, a small proportion of words will come after it when ranked alphabetically, thus the BST is highly unbalanced as almost every word entered after we is greater alphabetically. The AVL tree however, produces a balanced tree and consistent logarithmic run times.

**Space-Time**

As stated above, in general AVL trees will have more consistent run times and lower asymptotic run times due to their balanced nature. Additionally, the maximum height of an AVL tree is log(n) because of its rebalancing while a Binary search tree can have heights up to n. However, while the structure can differ greatly, both require the same amount of memory to save the trees as both save n nodes just in different locations. While both trees require the same amount of space for their nodes, AVL trees do require more operations to constantly rebalance the tree and thus need more memory to perform these operations.

**When to use AVL trees**

AVL trees are useful when creating structures that need to be searched a lot but don’t have a lot of insertions. Since each insertion on an AVL tree potentially takes multiple operations to rebalance the tree, it is less efficient in adding elements than a binary search tree. However, its asymptotic runtime of log(n) is quicker than the n runtime of a binary search tree. So, for things like databases that contain a lot of information that needs to be searched quickly but doesn’t need to be changed very frequently it is better to use an AVL tree.