Lab 9 & 10

Ember Roberts, Maggie Lyon, and Nyla Spencer

In this lab, we created a binary search tree that is able to automatically balance itself out within two levels of depth to increase efficiency. In a perfectly symmetrical binary search tree, each search cuts the number options in half, however, if the tree is not balanced, this is not the case. Take the case, for example, where items are only added in ascending order, and the structure of the binary tree ends up looking like a straight line rather than a split tree. In this situation, searching the tree is no more efficient than searching a linked list, as the number of search options are not actually cut in half each time we move down a leaf. As such, continuously balancing the tree allows us to maintain a big O of log2(n) for the find, insert, and remove functions, as every time they move down a level searching for an item or for a place to insert or remove an item, the number of possible locations is actually cut in half.

Contributions:

Ember – rotateLeft, rotateRight, node class

Maggie – Insert, remove, size, getAllAscending, main, Word class, balancing

Nyla – Find, getAllDecending, emptyTree