Author: Caleb Ewer

Date: 12/1/23

ID: 2409827

CS 260 Module 6 Journal

In this module we were able to work with a binary search tree. In my opinion I believe the binary search tree is the best used data structure we have learned in this class. This data structure works really well for our purposes because it can sort the bids into an easier structure. The binary search tree helps to lower the time it takes to access, modify, add, and delete elements.

The strengths of the binary search tree are faster insertion and deleting. For our purposes we could simply look up the bid id and work down the tree looking for the location or element to add or delete. Searched through the bids takes significantly less time than the other data structure has taken because instead of working through an entire list searching from the front or end of a list we can start from a root element do a comparison and move down that direction and repeat.

The weakness that a binary search tree has is that if it is unbalanced the time complexity is much worse. For our purposes we did not run into this problem but it is a real problem to consider when working through the programming aspects of it.

The binary search tree worked well in our application but there are several other applications we can use it for. One idea would be to use it in a spell checker. In a spell checker we could use the ASCII comparison for each letter and if its word is lower we store in the left and larger the right and just keep moving down the line. Because of the time it would take to search the word it would be very time efficient.